Contents lists available at ScienceDirect

Array

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

FaSIVA: Facial signature for identification, verification and authentication of persons

Elie Tagne Fute [a,b,*], Lionel Landry Sop Deffo [a,b], Emmanuel Tonye [c]

[a] Department of Computer Engineering, University of Buea, Buea, Cameroon
[b] Department of Mathematics and Computer Science, University of Dschang, Dschang, Cameroon
[c] Department of Electrical and Telecommunication Engineering, National Advanced School of Engineering, University of Yaounde I, Yaounde, Cameroon

ARTICLE INFO

Keywords:
Face recognition
Liveness detection
Low resolution image
Signature
Spoofing attack

ABSTRACT

The artificial reproduction of human vision capabilities has become unavoidable in recent technological applications, and as the number of applications increases, more challenges demanding specific attention in the proposition of solutions arise as well. Among this wide range of applications, there is face recognition in general and in video surveillance and access control where a lot of models have been proposed to suit users' needs. Although lot of progress has been made by proposing models for face recognition, but still there are issues affecting recognition accuracy. These issues include low resolution of input image and effective presence before capture equipment's. To address these issues, this paper presents a formal representation of face called face signature that considers three parameters namely, the quality of the input image, the patterns extracted from the image, and the characteristics of human presence namely eye blinking and liveness detection. Our experiment results show that the proposed signature is accurate and suits in real life applications where input images are of low resolution and spoofing attack are used.

1. Introduction

The recent migration to biometric technologies in most applications increases the interest in proposing solutions to monitor persons in an environment. Among these approaches are fingerprint-based control systems, iris-based control systems, face recognition and verification. However, the need to deploy autonomous systems focus our attention on face recognition because this category does not need interaction between the system and its users. One of the key conditions to achieve reliable face recognition is face representation. This is not always an easy task because of the variability of intra and inter-face characteristics. To illustrate how difficult it can be, let us consider a 3-D matrix representing a face. A single rotation of the given face gives a completely different matrix representation. So, the challenge in face representation is to find a description that will be invariant after rotation, translation, illumination, etc.

Many representations have arisen to this effect and one of the first was fractal representation [1]. This approach uses the theory that states an object can be represented by itself, that is, by auto similar elements. It was advantageous in the sense that it used fewer computational resources than the traditional principal component analysis (PCA) approach [2], however, the insertion of a new sample would lead to a recalculation of feature vectors. Many other representations have followed, and they are essentially based on the representation from a matrix (usually 3-D) to a 1-D feature vector. This is to reduce the use of computational resources. However, most of these representations assume that the image has a good resolution and do not handle spoofing attacks.

We propose, in this paper, a face representation to answer these challenges. It is a signature named FaSIVA which stands for **F**ace **S**ignature for **I**dentification, **V**erification and Authentication of persons. The idea is to propose a robust face representation that will not only handle the identification process but will take into consideration reinforcement processes such as image enhancement, verification, and authentication.

The rest of the paper is organized as follows; Section 2 presents previous work with an emphasis on outstanding work related to our study. In Section 3, the details of the FaSIVA signature are presented where all the formalization processes are explained. To validate the model, we present, in Section 4, the different implementations that have been carried out as well as the results obtained. The work is concluded in Section 5 with a summary and some thoughts for future work.

[*] Corresponding author at: Department of Mathematics and Computer Science, University of Dschang, Dschang, Cameroon.
E-mail address: eliefute@yahoo.fr (E. Tagne Fute).

https://doi.org/10.1016/j.array.2021.100112
Received 5 May 2021; Received in revised form 25 August 2021; Accepted 19 November 2021
Available online 3 December 2021

2. Related work

Earlier approaches used for face recognition where approaches usually denoted as hand engineered approaches which include local binary patterns (LPB), Fisher vectors, Scale Invariant Feature Transform (SIFT).

The principle used to represent a face is called sparsity which consists of finding minimum elements using the original matrix of the face image that can robustly and accurately represent the given face. The result obtained is called the features vector. A mathematical principle is usually used to find the given features vector. In this view, many approaches have been developed in the literature; some famous ones are local binary patterns (LBP), eigenfaces and fisher faces.

For LPB, the surrounding of each pixel is studied [3]. Here, a binary pattern is computed for each pixel using a mathematical formula and a threshold, and at the end, all the values computed are gathered to derive the image histogram. Since it uses a threshold mechanism, LPB is sensitive to local variations of illumination and does not perform well when there is partial occlusion.

While LPB focuses on local values of pixels to compute the features' values, eigenfaces [4] uses a projection on a new vector space with a lower dimension using a principle called principal component analysis [5]. This method is advantageous in that information related to geometry and reflectance of faces are not required and the recognition process is efficient. However, the approach is slow and to reach an acceptable level of accuracy, many sample images must be considered.

The case of Fisher faces [6] is similar to eigenfaces, however, they are different in that another principle called LDA (linear discriminant analysis) [7] is added to enable extraction of characteristics per class rather than considering the sample images as a whole when extracting eigenvalues. This enables the separation of one class from another. It has the advantage of reducing the error rate.

For SIFT features [8], most of related algorithms are composed of four part which are scale-space extrema detection used to detect blob structures in an image. At this stage, a scale space is constructed where the interest points are detected. Then we have unreliable key-points removal where each candidate key-point is evaluated, and a decision is made depending on its value (below a threshold). This is followed by orientation assignment where one or more orientations are assigned to a key-point based on local image gradient directions. Finally, we have the key-point descriptor where image gradient are sampled around the key-point location using the scale of the key-point to select the level of the Gaussian blur for the image.

More recently by the end of 2011, face representations based on neural network models are gaining the field due to the quality of the results obtained with these models. Many models exist to this effect, and two of the most popular are the FaceNet model [9] and visual geometric group (VGG) model [10]. Both models use neural networks with a certain number of layers. FaceNet has 22 layers and a triplet loss layer at the end which increases the distance between two samples of the same class and does the opposite on two samples of different classes during the training process. VGG, on the other hand, has many variances; VGG-16 [10], VGG-19 [11], ResNet-50 [12]. Other famous models exist such as the Inception [13] model and Alex network [14]. More investigation on these models is highlighted in [15]. Note that all these models use convolutional neural networks which are a specific type of network designed to take a 3-D (generally RGB) image as input.

Apart from the mentioned model many other model exist in the literature some of them include Deepid3 [16] is rebuilt from stacked convolution and inception layers proposed in VGG net and GoogLeNet to make them suitable to face recognition. Erjin et al. [17] propose a model called Megvii which achieves 99.5% accuracy on LFW benchmark. Also, there is also DeepID2+ [18] which is learned with the identification–verification supervisory signal. By increasing the dimension of hidden representations and adding supervision to early convolutional layers. Another model includes the model of Zhenyao et al. [19] which is a deep learning framework that can recover the canonical view of face images. It dramatically reduces the intra-person variances, while maintaining the inter-person discriminativeness. Moreover Xudong et al. [20] proposes a principled transfer learning approach for merging plentiful source-domain data with limited samples from some target domain of interest to create a classifier that ideally performs nearly as well as if rich target-domain data were present. Dong Chen et al. [21] introduces a Bayesian face recognition method which model two faces jointly with an appropriate prior on the face representation and had great results. Another popular model is DeepFace [22] which is trained on the large facial dataset of four million facial images belonging to more than 4000 identities. As consequence, it reaches an accuracy of 97.35% on the Labeled Faces in the Wild (LFW) dataset, reducing the error of the current state of the art by more than 27%.

2.1. Choice of convolutional neural network

As we already know convolutional neural networks are specific types of networks which their inputs are tailored to receive a 3-D matrix that an RGB (color image). It has the advantage that it handles images with different positions, adversarial examples, coordinate frame and sometimes for the performance. This is because it includes adding artificial offsets to the inputs, so that the network learn to cope with differences [23]. So, position information is implicitly encoded within the extracted feature map. Also initial layers are most often used as edge detectors so that each subsequent layer takes those as inputs and guesses higher and higher level concepts as you go deeper.

Concerning adversarial examples, they include situations in which an attack generates a patch that can be placed anywhere within the field of view of the classifier and causes the classifier to output a targeted class [24]. This leads the network to attain high confidence in the incorrect classification of an adversarial example and the capacity of fooling the system with imperceptibly little noise. To overcome this, a naive approach consists of pretending to be the attacker, generate several adversarial examples against your own network, and then explicitly train the model to not be fooled by them. Recently, more specific approaches have been proposed as an example defensive distillation method of Papernot et al. [25] which trains a distillation model of the same scale. An extensive study of such approaches is presented by Bingcai et al. in [26].

Finally to cope with coordinates in the frames CNN introduces coordinate independent features [27]. Since convolutional networks extract a hierarchy of feature fields from an input signal on a manifold, features are thereby computed via kernels, optimized to detect characteristic spatial patterns in lower level features. This give CNNs the capacity to easily handle coordinates when switching from on frame to another.

3. Description FaSIVA signature

3.1. Generalities

To design a complex algorithm, a top-down approach is taken and the analysis proceeds by successive refinements. By so doing, we stay away from any implementation, the concrete representation of data is not fixed; refers to the abstract data type. We give a notation that describes the data, the applicable operations to these data (primitives), and the properties of these operations (Semantic). According to [28], an abstract data type (ADT) is a mathematical specification of a set of data and a set of operations that can be performed on them. This type qualifies as abstract because it corresponds to a set of specifications that a data structure must then put in the work. It allows to define non-primitive data types which are not available (not already implemented) in current programming languages. However, the framework for formally defining the types and operations on objects of these types is that of algebraic specifications. More precisely, the semantics of an algebraic specification consists of the definition of one or many data which it describes. An algebraic specification, therefore, describes one or more data types through signature, and a set of axioms. The signature is a set of names of a type called sorts, and a set of operations in which each operation's name is associated to a profile of the form $S_1 \quad S_2 \ldots S_n \to S_0$ where $S_i = 0 \ldots n$ are sorts.

To define an abstract type, we will need certain fields which are

- **Introduction:** defines the (name of the type) and other specifications used.
- **Description(optional):** Informally describes such operations.
- **Operations:** defines the syntax of the operations in the interface and their parameters.
- **Axioms:** defines the semantics of operations in the form of equations. So, a specification that is a combination of spells and operations is equivalent to a signature, and the axioms must define the operations completely and correctly.

Note that the name sort is more specific than the name of the type. While a sort is a syntactic object, a type is both a set of data and a set of features working on these data [29]. Also, several different types can be associated with the same set of data according to the functionality considered. Moreover, the number of elements the function requires is called the arity [29]. The term function here represents any operation or any definition requiring operations.

In order to clearly present the FaSIVA signature, we initially present some basics concepts needed to define a signature in general. We continue by characterizing the signature itself by declining all its component then we derive from this an equational representation with the different metrics. Finally we explain how the proposed signature will be used in a form of an overall flowchart.

3.2. Definition of the facial signature

According to [30], a signature is defined by a set of sorts, S, and by a set, A, of operation names, each provided with an arity on S. The operation names must be distinct two by two. It is not required that the set A must be finite, although in practice it is always the case (if only to be able to write a specification in a finite number of characters).

It is important to highlight the fact that before proceeding to any representation of the face by a signature, we obviously need a cropped image from the input frame containing the face for that we need a face detection approach. In our case the approach used is MTCNN [31] which a cascade of three neural network to detect and align the face in a bounding box.

Let $I = M(i,j,k)(i,j,k) \in IN^3$ be a facial image, we want to find a representation, S of I such that this representation contains enough elements to uniquely represent a face so that this representation can be used to differentiate between two faces. At first glance, we would like to obtain a simplified representation that is a vector with fewer elements which will facilitate the processing. We already have the existence of a representation proposed by [9] which consists of representing a face by a vector of size 128 called an embedding, which is the basis of their FaceNet project. Interesting results have been obtained with this representation, that is why it has been and is used in most recognition approaches based on deep learning. However, this approach still requires a large amount of training data to be effective. In addition, a size 128 vector is used, giving the intuition that an increase in the size of this vector will lead to an improvement in the results, that is why we have focused our thinking on this track. Recently, a new approach is under study which involves representing a face with an embedding vector of size 512 and 2048 [32,33] with a corresponding neural network. These treatments are carried out using the assumption that the images are of good quality which is not always the case in real-life. In this perspective, [34] proposes a method allowing to improve the quality of the image, that is, to transform an image of small resolution to a high-resolution image. As we mentioned in the previous section, we opted for the use of the family of methods using the Resnet network which allows obtaining a vector with a size larger than 2000.

3.3. Equational representation of the signature

Before defining the new facial signature, it is important to situate the paradigm implemented. The idea is that for an image, I, we find a representation $f(I)$ such that;

$$f(I) = (P_1, P_2, \ldots, P_n) n \in IN \tag{1}$$

Here, the $P_i, i = 1, 2, \ldots, n$ are characteristic elements which can be vectors or real parameters. This is to find a representation that uniquely characterizes a face. Therefore, the more parameters we have, the more precise will be the representation. Moreover, in real-life, the processed images are still not of good quality as is often the case in datasets that presuppose that the images are of good quality. This, therefore, introduces new sets of constraints that can be related to illumination, rotation, resolution, occlusion, etc. As we cannot tackle all these issues simultaneously, we have chosen the ones which are convergent, including illumination, rotation, and resolution. Consequently, we represent a face by a vector of invariant characteristics by any operation affecting the texture (illumination, etc.) and geometric transformation (rotation, translation, etc.), and introduce a parameter which is a function of the image quality such as the resolution.

We define our new signature as follows: Let $I = M(i,j,k)_{(i,j,k) \in IN^3}$ a facial image, we extract a signature S.

$$S(I) = (R, F, E, A)$$

$$\text{with} \begin{cases} R = (h, w), & \text{the super resolution vector} \\ F = (f_1, f_2, \ldots, f_{2062}), & \text{the identification vector} \\ E = (e_1, e_2, \ldots, e_{128}), & \text{the verification vector} \\ A = (a_1, a_2) & \text{the authentication vector} \end{cases} \tag{2}$$

Super-resolution is a principle that allows the reconstruction of an image using the formula

$$F(Y) = \omega_3 \times F_2(Y) + B_3 \tag{3}$$

with

$$F_2(Y) = f(\omega_2 \times F_1(Y) + B_2) \tag{4}$$

and

$$F_1(Y) = (\omega_1 \times Y + B_3) \tag{5}$$

In these formulas Y is a low-resolution image, $\omega_3; \omega_2$ and ω_3 the kernels of the successive convolutions and B_1, B_2 and B_3 the biases, finally, $F(Y)$ is the reconstituted image. The identification vector is obtained using the Resnet-50 model [12] which extracts the features on the input image that must be of size 224. The verification vector is obtained using the FaceNet model which extracts the characteristics on the input image using the "triplet loss" algorithm [35] expressed by:

$$L = \sum_{i=1}^{N} \left[\left\| f(x_i{}^a) - f(x_i{}^p) \right\|_2^2 - \left\| f(x_i{}^a) - f(x_i{}^n) \right\|_2^2 + \alpha \right] \tag{6}$$

Where $x_i{}^a$ is the anchoring image of a specific person, $x_i{}^p$ a positive image of the same person and $x_i{}^n$ a negative image. A is obtained using two methods, the first one consists of calculating the acceptance coefficient taking into account the radius light incident on the facial surface as follows;

$$I(x, y) = f_c(x, y) \times \rho(x, y) \times A_{light} \times cos\theta \tag{7}$$

While $cos\theta$ is the cosine between the normal surface and the incident radius, ρ is the reflection coefficient.

The second method consists of using the number of eye movements which is calculated using the formula of [36]. To do this, each eye is represented by 6 points of coordinates (x, y)

$$EAR = \frac{\left\| p_2 - p_6 \right\| - \left\| p_3 - p_5 \right\|}{2 \left\| p_1 - p_4 \right\|} \tag{8}$$

Where $p_1, \ldots, p_6$ are 2D landmarks. EAR is called "Eye Aspect Ratio" which is the ratio of appearance of the eye. The numerator of this equation calculates the distance between the vertical landmarks while the denominator calculates the distance between horizontal reference points. The weighting of the denominator is because there is only one set of horizontal points but two sets of vertical points. The use of this equation avoids the techniques of image processing and simply depends on the ratio of the distances of the points of marker to determine if a person is blinking. Thus;

$$S = (w, h, f_1, f_2, \ldots, f_{2062}, e_1, e_2, \ldots, e_{128}, a_1, a_2) \tag{9}$$

3.4. Use of the defined signature

We apply the following steps to an input image I:

- **First step:** Determination and comparison of the resolution. Here, the resolution of the image is calculated and if it is lower than the threshold resolution R_s, the super-resolution principle will be applied to it to increase the said resolution. This will generate a new image with a resolution greater than or equal to R_s.
- **Second step:** Extraction and use of the first features vector F. This step entails extracting the features F vector from image I, and then using the SoftMax regression layer to determine its closest vector. Thus, this process results in the claimed identity of the individual.
- **Third 3:** Extraction and use of the second features vector E. Once the allegedly identified individual is identified, the process is reinforced by a verification or authentication step that will involve verifying that the individual is indeed who he claims to be. For this, the second features vector E is extracted from image I and compared to the one present in the knowledge base using Euclidean distance. This verification will produce a more confident result.
- **Fourth step:** Extraction and use of the third features vector A. Now that the individual is identified and the identification is strengthened by a verification (first part of the authentication), we proceed with the extraction of the third features vector A, which will be used to verify the characteristics of human presence (second part of the authentication) as described in Section 4.3. So, if all these checks are correct, one can indeed conclude that it is a real person and not a spoofed person that has been identified.

These steps are represented on the general flow chart of the signature presented on Fig. 1

4. Implementation and results

FaSIVA signature has three parts namely the super-resolution part, the features extraction part, and the authentication part. We are, therefore, going to show, in this section, the results obtained in each part while emphasizing the advantages of the results over previous ones.

4.1. Super resolution part

This essentially comprises the enhancement of image quality by increasing its resolution in case it is less than the threshold resolution. The architecture used for super-resolution is presented in Fig. 2.

4.1.1. Functioning principle of the module

Since in real-life scenarios it is not always evident to have images of good quality (that is, of good resolution), the resulting processing will be altered and the results negatively influenced, leading to poor or incomplete features extraction. To cope with this situation, we introduce a super-resolution module that enables us to increase the resolution of the input face when it is below a certain threshold determined empirically. This approach uses a convolutional neural network that takes an image of small resolution as input and reconstitutes an image of higher resolution. Many approaches exist in the literature, but we have chosen a famous and recent one, the fast super resolution convolutional neural network (FSRCNN), based on its results. FSRCNN enhances images of objects by multiplying their resolutions with a positive factor k, and we have adapted it to enhance human faces with a dataset that we constructed.

4.1.2. Dataset construction for training

The usual approach entails using a dataset of random images to train the model in as much as the images are of good quality (resolution). However, this model performs poorly in the specific case of human faces, that is why we decided to train our model specifically on face images. To do so, we replace the dataset of objects with human faces. In our case, the faces have been retrieved in the label faces in the wild (LFW) dataset because it possesses enough images, that is, 13233 images that we have separated into two sets with a ratio of 3:1, for training and testing, respectively. This gave us a set of 9924 images for training and 3309 images for testing. The model was then trained on the dataset with 20 epochs.

The fundamental idea behind this network usage is to reconstitute an image of better resolution by the intuition developed by convolutional neural networks. The resulting process differs from a simple zoom because its algorithm adapts values of the image matrix on a new dimension which generally conduce to loss of certain feature elements. On the other hand, the super-resolution approach uses an initial image of little resolution and reconstitutes a similar image of better resolution. It, therefore, highlights hidden details enhancing in this case, features extraction.

4.1.3. Impact of super resolution on face detection

To illustrate the impact of super-resolution on face detection, we train a model that multiplies the resolution of the input image by a factor k. The evolution of the loss function based on the number of epochs is presented in Fig. 3. With this model, enhancement of input image quality is assured and our factor k is equal to 4.

Note that the value 4 has been chosen empirically. For this purpose we tested different values of the factor starting at 1.5, then we tested on the following values 2, 2.5, 3, 3.5 and 4 value at which we started having significant results. It is for that reason we stopped there, whereas this values can be changed depending on the type of application we are using.

After that, we use 200 images of 20 individuals with a ratio of 20 images per individual. Many copies of these images are made and downscaled on the following resolutions; $25\times25, 35\times35, 45\times45, 55\times55$. The idea is to illustrate that the super-resolution principle increases the performance of face detection. Consequently, the summary of images detected as well as their corresponding percentages are reported in Table 1. We notice that in each case, the number of faces detected with super-resolution is greater than the number detected without. We also notice that the higher the input image resolution, the better the output image after the super-resolution and an acceptable image resolution is 35 × 35. This is what will be used as our threshold resolution in the presented process.

4.2. Facial features extraction part

This module is essentially dedicated to features extraction on input images enabling unique representation of each one. We have seen in the literature that many models exist but two have captured our attention: FaceNet, VGG and Resnet. This choice is motivated by the ratio between the results obtained and the time taken. Consequently, we did tests on the LFW with three models FaceNet, VGG and ResNet-50 the results obtained are presented in Table 2

The VGG model uses an RGB image as an input of fixed size 224 × 224. This image later passes through many convolutional layers

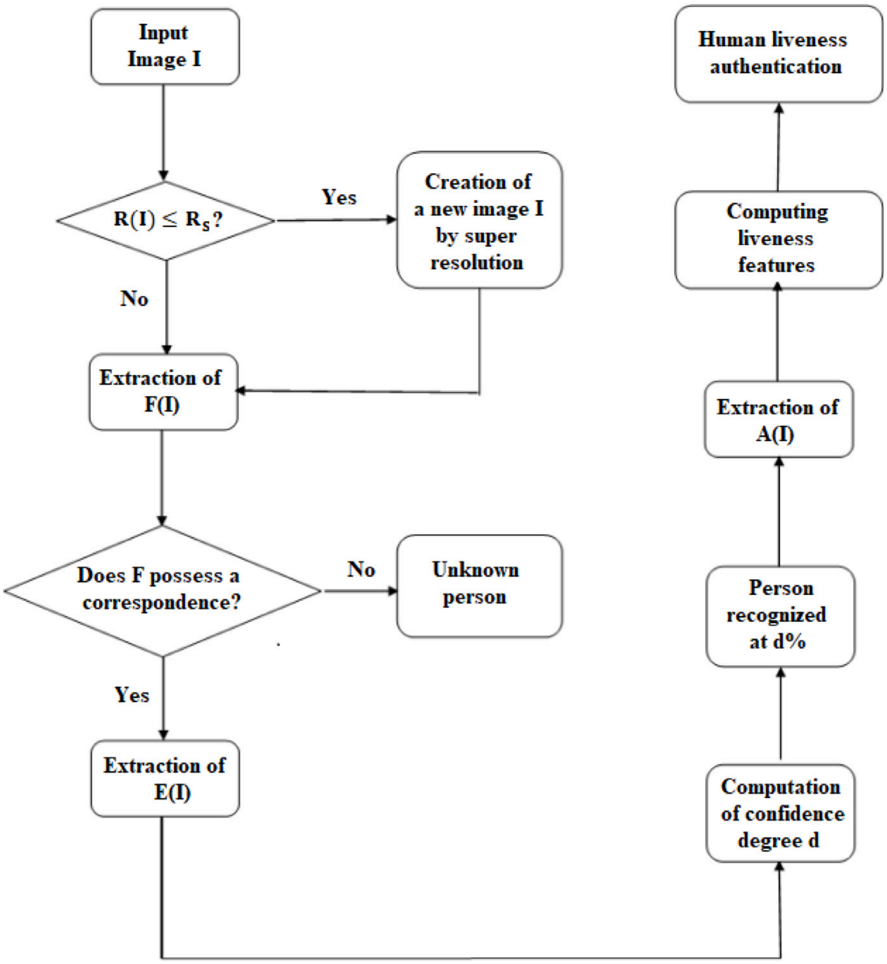
Input Image I
R(I) ≤ R_S ?
Yes
Creation of a new image I by super resolution
No
Extraction of F(I)
Does F possess a correspondence?
No
Unknown person
Yes
Extraction of E(I)
Computation of confidence degree d
Person recognized at d%
Extraction of A(I)
Computing liveness features
Human liveness authentication

Fig. 1. General flow chart of FaSIVA signature.

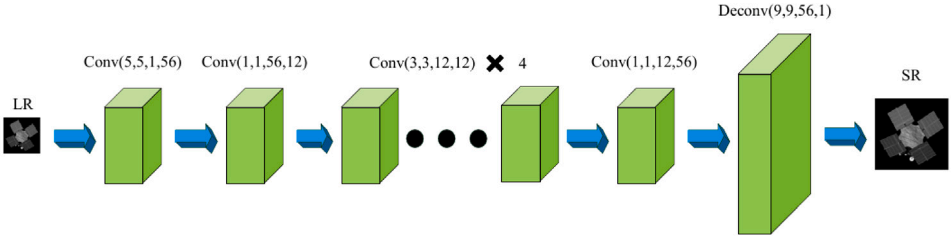

Fig. 2. Architecture FSRCNN network.

Table 1
Number of detected faces in function of input image resolution.

Résolution	25 × 25	35 × 35	45 × 45	55 × 55
Faces detected without super resolution (on 200)	175	189	189	191
Detection percentage in %	87.5	94.5	94.5	95.5
Faces detected with super resolution (on 200)	191	195	195	197
Detection percentage in %	95.5	97.5	97.5	98.5

Table 2
Comparative study of models used in features extraction.

Type of the network	Number of images	Number of detected faces	Size of features vector	Accuracy of recognition in %
FaceNet	3243	2610	128	79
VGG	3243	2610	512	80
Resnet-50	3243	2610	2048	95

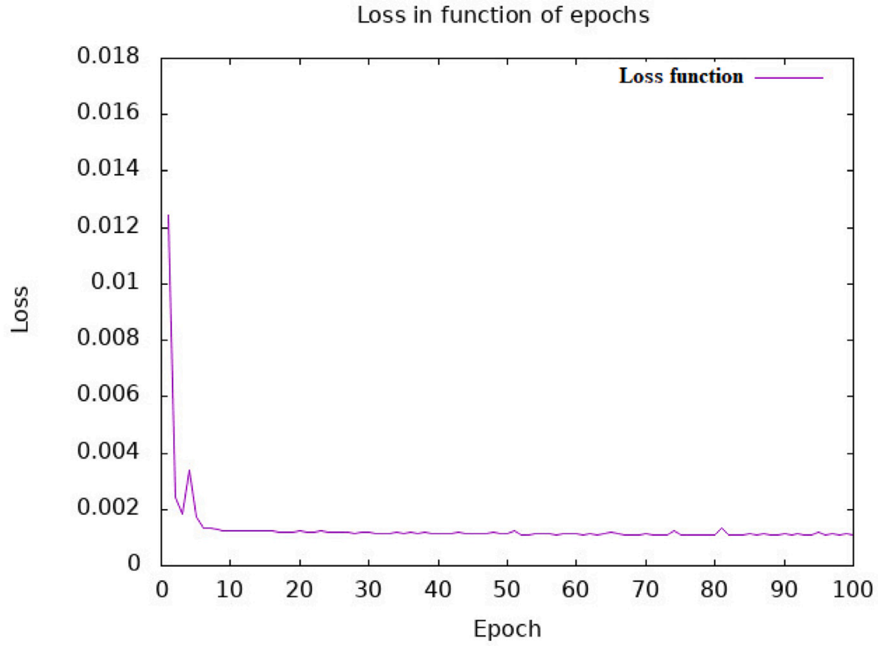

Fig. 3. Evolution of loss function during super resolution model training.

where filters have been used with a small reception field of 3 × 3. Note that this is the smallest value to capture the notion of up/down, left/right, center. In one of its configurations, it also uses convolutional filters of 1 × 1 that can be seen as linear transformations of input channels. The spatial pooling is assured by 4 layers of maximum pooling that follow certain convolutional layers on a 2 × 2 frame with a step of 2. This model has been trained for weeks on a Titan with a graphical processing unit (GPU) of 837 Mhz frequency, 3004 Mhz of memory frequency, and a Graphics Double Data Rate version 5 (GDDR5) type of memory.

4.2.1. Facial recognition part

This is the easiest step once the features have been extracted. It involves matching a features vector of an input image to a features vector belonging to a class in the knowledge base. The native approach used to this effect is the KNN (K-nearest neighbor) algorithm which computes the distance between each input image features vector and all images in the knowledge base and returns the class with the lowest distance. Another approach is the use of a support vector machine (SVM) trained to efficiently separate data for further use that can be either classification or regression. Even if its execution time is negligible once the model has been trained, its performance decreases considerably when the number of classes increases or when data are not linearly separable. We, therefore, choose to use this approach which is the use of a trained softmax layer that takes into consideration details of the features vector and uses less time for execution as well as SVM.

4.2.2. Comparative study of KNN and softmax complexity

Let n be the number of features vectors and m the number of classes (that is the number of individuals). We admit here that we have approximately n individuals per class. To study the complexity of both approaches, let us look at the following algorithms:

Algorithm 1 presents the execution of the KNN approach in classification.

- If we have 1 individual we will have $n \times 1$ computations, and so, a complexity of $\theta(n)$
- If we have 2 individuals we will have $n \times 2$ computations, and so, a complexity of $\theta(2n)$

Algorithm 1: Use of KNN for classification

```
Input: Features vector $v$ of input image
Output: Label of identified individual
Data: Set of known features vector
Data: m: number of classes; n: number of vectors per class
Initialization for each $c_i$ class in m do
    for each $v_j$ vector in $n$ do
        compare $v_j$ and $v$
if correspondence then
    return the label of the corresponding individual
else
    return the label of the closest individual
```

- If we have 3 individuals we will have $n \times 3$ computations, and so, a complexity of $\theta(3n)$
- etc
- If we have m individuals we will have $n \times m$ computations, and so, a complexity of $\theta(nm)$
- Finally, if $n \ll m$ which is generally the case in real-life, we will have a complexity greater than $\theta(n^2)$

This makes its execution difficult in real-life scenarios, that is why we prefer to use a trained softmax layer because once the layer has been trained, it directly outputs the label of the corresponding individual as can be noticed in Algorithm 2.

Algorithm 2: Use of softmax for classification

```
Input: Features vector $v$ of input image
Output: Label of identified individual
Data: Trained softmax layer
if correspondence then
    return the label of the corresponding individual
else
    return the label of the closest individual
```

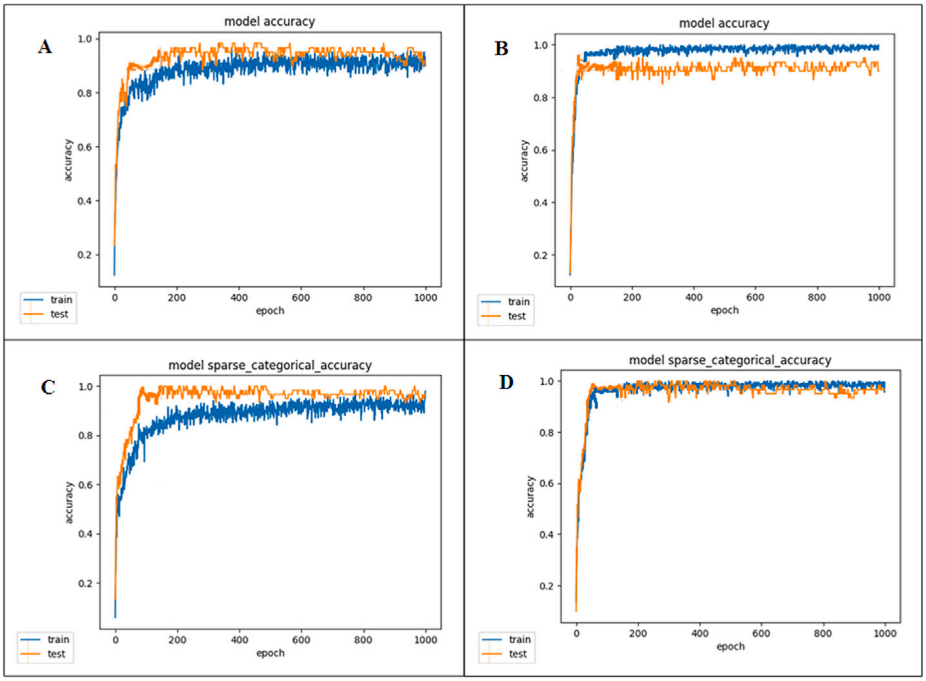

Fig. 4. Evolution of precision during training of softmax layer; A: Softmax layer, B: Softmax + large margin, C: Softmax + CRelu activation, D: Softmax + large margin and CRelu activation.

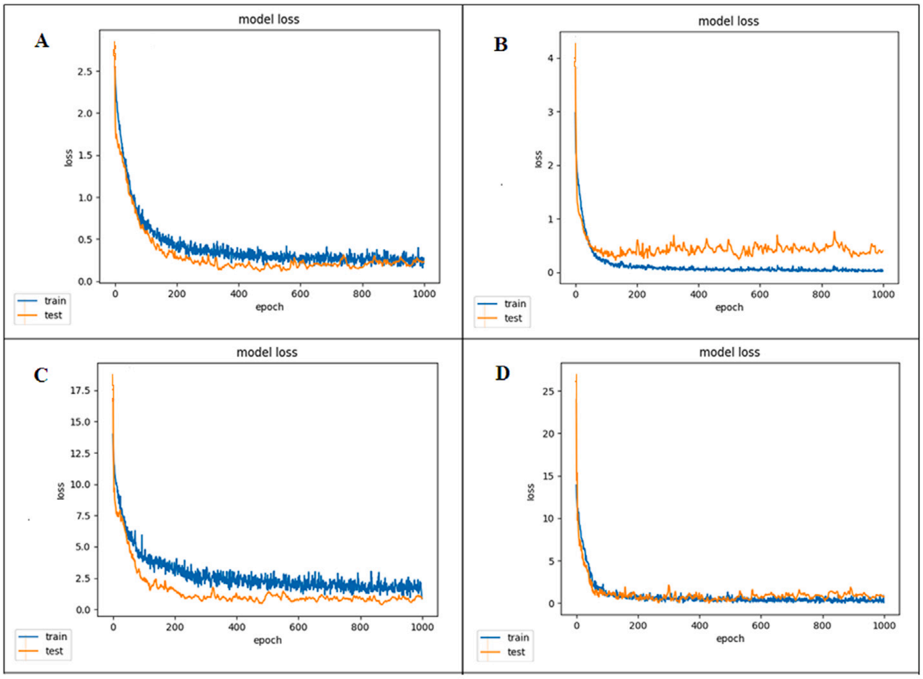

Fig. 5. Evolution of loss during training of softmax layer; A: Softmax layer, B: Softmax + large margin, C: Softmax + CRelu activation, D: Softmax + large margin and CRelu activation.

We notice that just one computation is needed, leading to a complexity of $\theta(\epsilon)$ where ϵ is the instruction complexity with $\epsilon \ll n$.

4.2.3. Impact on training and validation

We have proven in [37] that an introduction of margin layer increases classification capacity, and this can be reinforced by concatenated rectifier linear unit (CReLU) activation function as demonstrated in [38]. We have, therefore, experienced four scenarios; the first is the training softmax layer with normal situation and parameters, the second is the training with the introduction of the margin layer, the third is the training with the use of the CRelu module as the activation function, and the fourth is the combination of the previous two. Fig. 4 and Fig. 5 respectively present accuracy and loss evolution during training and validation steps. We can easily notice a better stabilization of our combined approach in the two cases (evolution of precision and dropping of loss).

Table 3
Mean recognition percentage in function of image resolution.

Resolution	25 × 25	35 × 35	45 × 45	55 × 55
Normal recognition (in%)	5	36	63.5	79
Recognition with super resolution (in%)	32.5	58	76	88

Table 4
Summary table of images per category for training.

Dataset	NUAA	Replay-attack	Casia	Total
Fake	1748	1504	1669	4921
Real	1743	1283	1618	4644
Total	3491	2787	3287	9565

Note that we extract the softmax layer of the chosen ResNet-50 model, load the pre-trained weights from [12] which focuses on the extraction of facial features. We then add a softmax regressor which was trained on our own dataset. To illustrate our methodology, we used 400 images of the previous 20 individuals with a ratio of 20 images per individual to train the given layer and a total of 60 images with a ratio of 3 images per individual for validation.

Since we were trying to illustrate the impact of the introduction of CReLU activation function as well as the insertion of margin layer we did not recorded the confusion matrices in different cases, we instead focused our attention on the evolution of training accuracy and loss which is more illustrative in our case.

Thereafter, we use 200 images of these 20 individuals with a ratio of 10 images per individual. Many copies of these images are made then each copy is downscaled for the following image resolution $25 \times 25, 35 \times 35, 45 \times 45, 55 \times 55$. The idea is to show how super-resolution increases face recognition performances. Table 3 presents recognition percentages as a function of input image resolution.

If the images contain some degree of tilt or rotation the network model chosen which is Resnet-50 model is used. This model handles any rotation, pose and age variation. It was downloaded then pre-trained, and after a fine-tuning process is done on our data set and a fully connected layer for feature extraction is added. Consequently, the features extracted are independent of transformations.

4.3. Authentication part

4.3.1. Functioning principle

Formally, authentication comprises the verification that an individual is effectively the one he claims to be. In the frame of video surveillance, we will understand authentication to be any mechanism targeting the attestation of effective presence of the identified individual as well as any reinforcement decision-making principle. To do this, we first proceed to a verification of information. Concretely, we use features extracted by the FaceNet network, compute the distance (Euclidean) between this vector and the feature vector of the individual in the knowledge base. Note that a feature vector has previously been computed for everyone in the knowledge base. If the computed difference is less than the given threshold s, then the second phase can be performed. This phase involves liveness verification. A convolutional neural network is trained to this effect, to differentiate a real individual from a spoofed one. This spoofed individual can be an image or a video of a known person in the knowledge base.

The architecture of the network is presented in Fig. 7. It is an architecture inspired by the Alex network where pooling layers have been modified so that it can take a low image dimension (64 in our case) to increase the execution speed. Also, rather than training on one dataset, three famous datasets have been used namely, NUAA (Nanjing University of Aeronautics and Astronautics) dataset, Replay attack dataset and Casia dataset. This led to the increase of network generalization capacity and even outperforms one recent model with a very deep architecture. The choice of Alex network is motivated by the work of [39] which focuses on the sizes of the different kernels, compared to [40] which gained all its efficiency from its number of layers. We go in the same direction by also focusing on the dataset used which is why to increase the generalization capacity, we use a combination of several datasets. Moreover, this choice was also motivated by the training time and the execution time which are measured according to the number of frames per second; this will be discussed in the following section.

Finally, a third principle is applied which entails the detection of eye blinks. We use the principle stating that a person cannot stay for a certain amount of time in front of the multimedia sensor without blinking his/her eyes. Consequently, every image that does not contain one or more eye blinks within a given amount of time will be considered as a spoofed image or video. To achieve this, the landmarks approach is used to localize opened/closed eyes, so, by counting the number of blinks, we are done. Also, we extend the normal principle stating that eyes blink together by considering the blink of each eye separately because one can be injured on one eye or cannot have both eyes at all. Furthermore, we need to insist on the fact that this principle is applied if and only if the face has previously been detected as genuine. This gives rise to the flow chart in Fig. 6 illustrating how the proposed blinking principle works compared to the native one.

4.3.2. Training and testing

To increase the generalization capacity of the network, we used as previously mentioned, three famous data sets which are the NUAA, Replay attack and Casia datasets. We constructed our dataset as follows:

- **First step:** The NUAA dataset has 1748 fake images and 1743 real images in the training set, and 5761 fake images and 3362 real images in the testing set, so, no additional action was needed.
- **Second step:** The Replay attack dataset has in the training set, 300 fake videos and 60 real videos, and has in the testing set, 400 fake videos and 80 real videos. So, to extract frames, we had to take into consideration overfitting avoidance and image balancing during training, which is why we extracted 6 images per fake video and 25 per real video. This gave us a total of 1800 fake images and 1500 real images. These images were later passed to our face detection system to extract faces per image. The approach used here was the MTCNN (multi task cascade neural network) approach, and we had as result, a total of 1504 fake images and 1283 real images. In the test set, we extracted all frames per video given and handed them over to the face detector which gave a total of 95881 fake images and 30493 real images.
- **Third step:** The Casia dataset has in the training set, 160 fake videos and 80 real videos, and 240 fake videos and 120 real videos in the testing set. Proceeding the same way, we extracted 11 images per fake video and 21 per real video. This gave a total of 1760 fake images and 1680 real images. These images were later passed to our face detection system to extract faces per image. We had as result, a total of 1669 fake images and 1618 real images. Similarly, in the test set, by extracting all the frames and passing them to the face detector, we had a total of 42704 fake images and 20870 real images.
- **Fourth step:** Finally, we combined all the images in four categories; train set fake, train set real, test set fake and test set real. The images of the train set are used during training with a ratio of 3:1 which means that 75% is used for the training itself, and the remaining 25% for validation.

The summary of different numbers of images per category is presented in Tables 4 and 5. Also, the evolution of accuracy and loss during training is presented in Fig. 8.

Fig. 8 presents the evolution of parameters in four scenarios of our implementation. Firstly, we investigated to find a very accurate

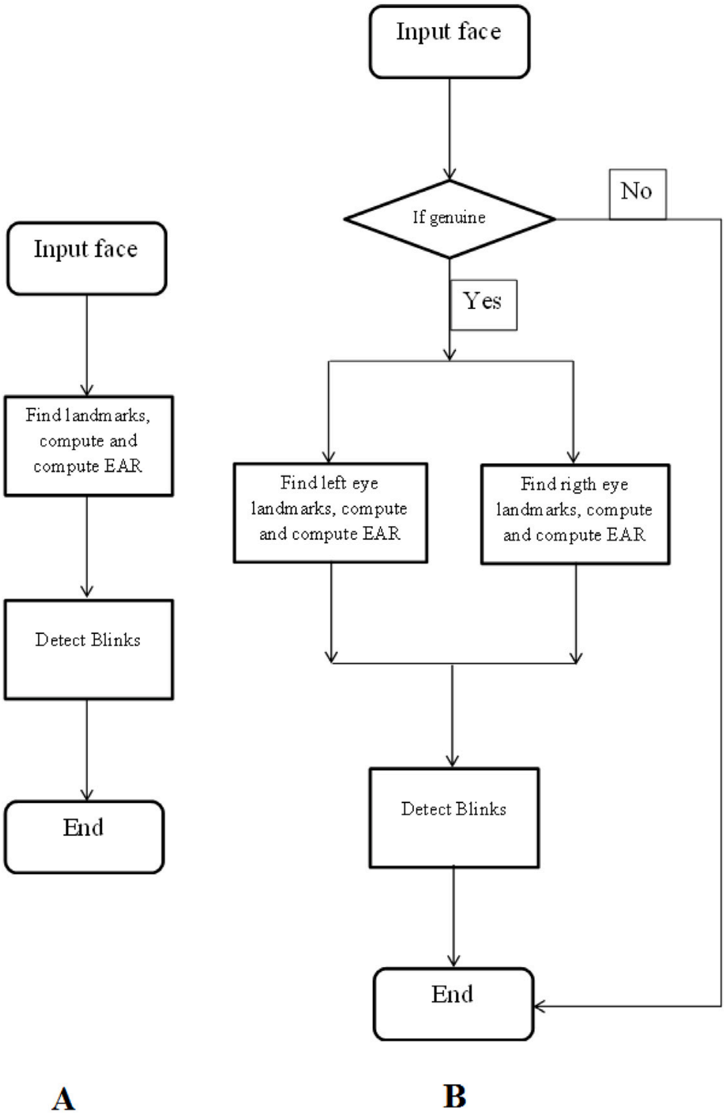
Input face
Find landmarks, compute and compute EAR
Detect Blinks
End
Input face
If genuine
No
Yes
Find left eye landmarks, compute and compute EAR
Find rigth eye landmarks, compute and compute EAR
Detect Blinks
End
A
B

Fig. 6. Flow charts of blink detection principle. A:Normal blink detection B:Proposed blink detection.

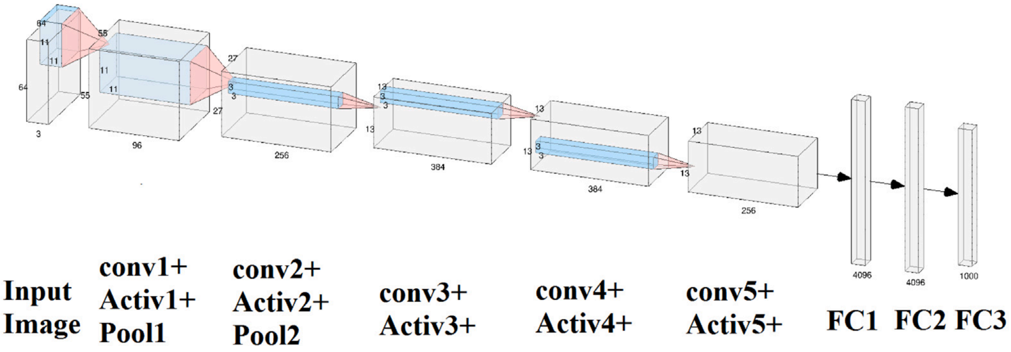

Fig. 7. Liveness detection architecture.

model, and compared to those found in the literature, we chose the Google Inception model [40], which in their work, has reached a test accuracy of 100% on the NUAA dataset with just 40 at the training phase. However, this model takes a lot of time to be trained, and due to its number of layers, it is a little slow during execution in real-time (see Table 6). We saw that adapting the convolutional kernel to reduce the input image size had a great impact on performance in the sense that it considerably reduces training time while conserving an acceptable accuracy, even though the execution time in real-life is still considerable. That is why we investigated once more and found that a less deep network can do the job, hence, we took the Alex network and adapted the convolutional padding for it to take an image of 64 × 64. Consequently, to avoid discarding any overlapping while striding with convolution or pooling kernels, each padding has been set to the same

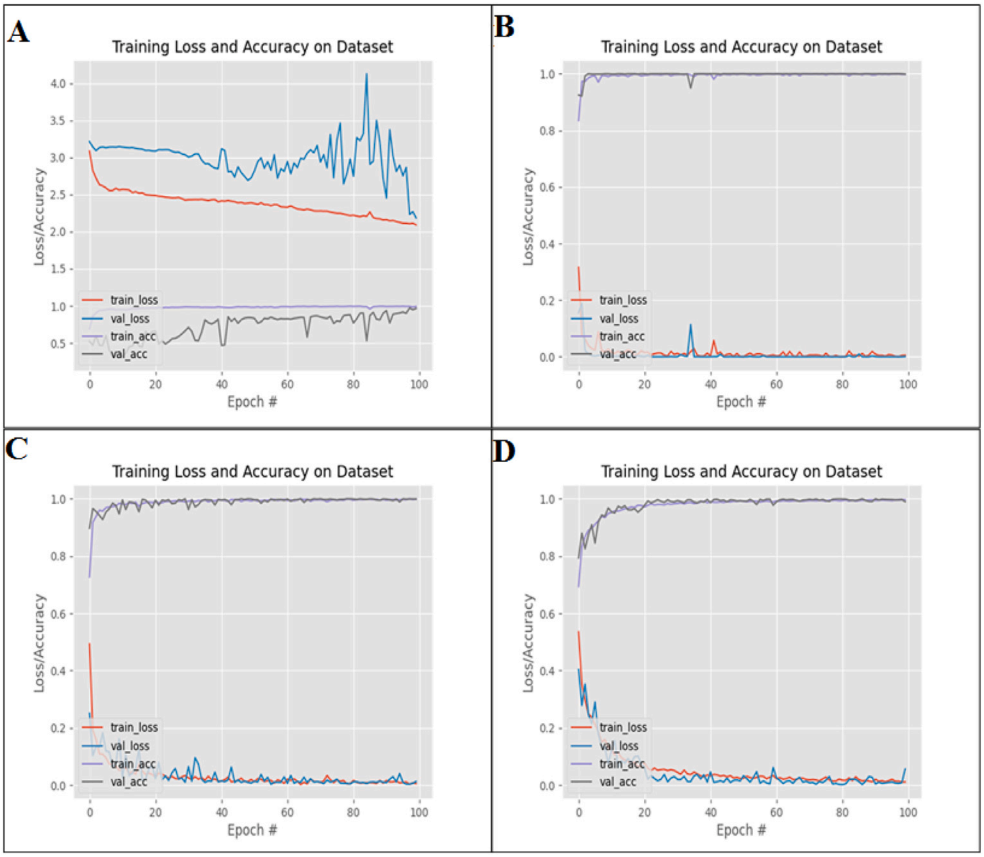

Fig. 8. Evolution of accuracy and loss during training phases in four scenarios; (A) Training using Inception model and NUAA dataset, (B) Training using adapted Alex model and NUAA dataset, (C) Training using adapted Alex model, NUAA and Replay dataset, (D) Training using adapted Alex model, NUAA, Replay and Casia dataset.

Table 5
Summary table of images per category for testing.

Dataset	NUAA	Replay-attack	Casia	Total
Fake	5761	95881	42704	144346
Real	3362	30493	20870	54725
Total	9123	126374	63574	199071

rather than valid. This process has progressively been reinforced by the Replay attack and Casia datasets. The results of different tests after training are presented in Table 7. Also, the training time and the execution frequency (frame per second) of each method is recorded in Table 6.

Note that in the Inception model, the authors in [40] have already adapted the convolutional and pooling kernels to switch from 224 × 224 to 64 × 64 that is why we were able to train the model which took 4 h 45 min on the NUAA dataset. The original Alex on the other hand, took 4 h 56 min to be trained on the same NUAA dataset reason why we adapted the input to also take 64 × 64 resolution images inducing a drop-in training time on the NUAA dataset giving a new time of 1 hour 51 min. This is the reason why we were able to train on a bigger and varied dataset, so, we first trained on a combination of NUAA and Replay datasets and later, we trained on a combination of NUAA, Replay and Casia datasets which gave training times of 3 h 12 min and 4 h 49 min, respectively.

In the case of execution time in terms of frames per second, we can easily notice that the proposed method outperforms the original Inception and Alex models with attention made on method 3 where the execution frequency is the highest. This can be understood because the more we increase generalization capacity, the lower will be the time to use the generated model.

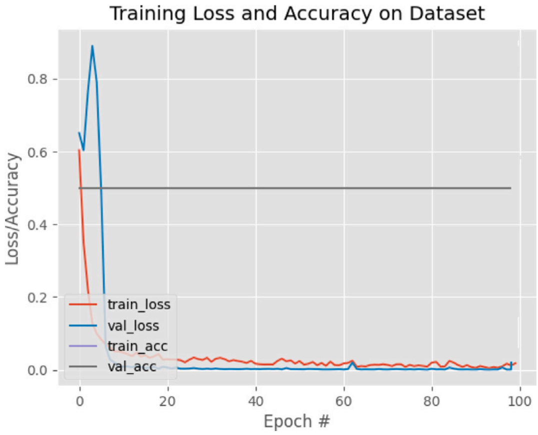

Fig. 9. Evolution of accuracy and loss during training phase with Resnet-50 model.

We can see that, with a lighter architecture as Alex network, we were able to achieve good performance compare to a deeper one such as Google Inception architecture. However, recent models exist in the literature were one of the most common is Resnet family architecture but due to their depth there are not yet widely used in liveness detection because usually and architecture with less layer is sufficient to have good results. Nevertheless, we train our dataset on one of it variant which is the Resnet-50 [12] architecture and the results obtained during training is presented on Fig. 9.

As we can observe, the results are quite similar with that obtained when using Alex architecture and combined data set for loss decrease, so no need to adopt the architecture because the results obtained with

Table 6
Training time and execution frequency of each method.

Method and Dataset	Training time in hours and minutes	Execution frequency in fps
Inception with NUAA	4 h 45 min	8.90
Alex Original with NUAA	4 h 56 min	8.30
Method 1: Modified Alex with NUAA	1 hour 51 min	9.70
Method 2: Modified Alex with NUAA and Replay	3 h 12 min	9.74
Method 3: Modified Alex with NUAA, Replay and Casia	4 h 49 min	10.4

Table 7
Results of different test scenarios; Inception with NUAA, Alex with NUAA, Method 1: Modified Alex with NUAA, Method 2: Modified Alex with NUAA and Replay, Method 3: Modified Alex with NUAA, Replay and Casia.

Approach	Dataset	TP	FP	TN	FN	FAR	FRR	TPR	TNR	ACC
Inception model	NUAA	3025	337	5263	173	0.0602	0.0541	0.9459	0.9398	0.9420
	Replay Attack	23725	6768	92247	3634	0.0684	0.1328	0.8672	0.9316	0.9177
	Casia	18942	198	40007	2697	0.0460	0.1246	0.8754	0.9540	0.9273
Alex original	NUAA	3362	0	5430	6	0.000	0.0018	0.9982	1	0.9993
	Replay Attack	29686	807	90499	5382	0.0088	0.1535	0.8465	0.9912	0.9510
	Casia	20568	302	42596	108	0.0070	0.0052	0.9948	0.9930	0.9936
Method 1: Modified Alex with NUAA	NUAA	3362	0	5423	13	0	0.0039	0.9961	1	0.9985
	Replay Attack	29491	1002	91729	4152	0.0108	0.1234	0.8766	0.9892	0.9592
	Casia	20655	215	42633	71	0.0050	0.0034	0.9966	0.9950	0.9955
Method 2: Modified Alex with NUAA and Replay	NUAA	3362	0	5435	1	0	0.0003	0.9997	1	0.9999
	Replay Attack	30492	1	95562	319	0	0.0104	0.9896	1	0.9975
	Casia	20669	201	42700	4	0.0047	0.0002	0.9998	0.9953	0.9968
Method 3: Modified Alex with NUAA, Replay and Casia	NUAA	3362	0	5432	4	0	0.0012	0.9988	1	0.9995
	Replay Attack	30493	0	95839	42	0	0.0014	0.9986	1	0.9997
	Casia	20757	113	42701	3	0.0026	0.0001	0.9999	0.9974	0.9982

Alex network are sufficient and satisfactory so no need to take a deeper architecture which will add the number of parameters and can impact on the system performance. However, the validation accuracy is worse around 50%, another reason to ignore the model.

The metrics used to evaluate the quality of the proposed approach are false acceptance rate (FAR), false rejection rate (FRR), true positive rate (TPR), true negative rate (TNR) and accuracy (ACC). These metrics are functions of the following parameters: true positive (TP), false positive (FP), true negative (TN) and false negative (FN) with the following formulas [36]:

$$FAR = \frac{FP}{FP + TN} \tag{10}$$

$$FRR = \frac{FN}{FN + TP} \tag{11}$$

$$TPR = \frac{TP}{TP + FN} \tag{12}$$

$$TNR = \frac{TN}{TN + FP} \tag{13}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

Note that these metrics indicate how good an approach could be. While FAR and FRR focus on the capacity of the approach to reject positive samples and should equal to 0 in a perfect scenario, TPR and TNR on the other hand focus on the ability of the model to accept true samples. This should be equal to 1 in a perfect scenario. We can easily see that our proposed approach outperforms the inception model in all the cases. Also, the more we combine datasets, the better the results in terms of accuracy; this is highlighted in Fig. 10 where we have successfully tested the NUAA dataset, Replay dataset and Casia dataset on Inception train with NUAA, native Alex train with NUAA, Alex model train only with NUAA, train with NUAA and Replay and train with NUAA, Replay and Casia.

Another name of TPR is Sensitivity. It is the proportion of true positives that are correctly identified by a diagnostic test. It shows how good the test is at detecting a disease. Also another name to TNR is Specificity which is the proportion of the true negatives correctly identified by a diagnostic test. It suggests how good the test is at identifying normal (negative) condition. Finally Accuracy is the proportion of true results, either true positive or true negative, in a population. It measures the degree of veracity of a diagnostic test on a condition.

Finally, to detect eye blink, landmarks points method proposed by [34] was used. This uses the following formula to compute if an eye is opened or not:

$$EAR = \frac{\|p_2 - p_6\| - \|p_3 - p_5\|}{2\,\|p_1 - p_4\|} \tag{15}$$

Where p_i are landmarks points as presented in Fig. 11

This last principle enables the reinforcement of liveness detection in the case of video input frames. So, if after a certain period (say x seconds), we do not detect a blink, we, therefore, conclude that we do not have a real person in front of the multimedia sensor. The number of seconds is detected empirically and according to the type of application we intend to develop. Figs. 12 and 13 illustrate how we have applied the formula to detect a blink on our input image with the normal blinking process and with the proposed blinking process. As we indicated previously, we do not rely on the strict application of its formula, instead, we first check if the face is genuine or spoofed. If it is genuine, rather than considering both eyes blinking together, we consider them separately. So, while computing eye aspect ratios (ear), if any of the eyes changes its ear value, we consider it to be a total blink.

It can be noticed that in the proposed approach, we can detect a blink when one eye has an eye aspect ratio less than the threshold aspect ratio (Fig. 13 Line 3) which is not the case in the original approach that focuses on two eyes blinking (Fig. 13 Line 2). On the other hand, in case both eyes have an eye aspect ratio greater or equal to the threshold aspect ratio, blink detection is achieved by the approaches without any problem (Fig. 12 Line 2 and 3). The input images used are frames taken from the talking face dataset [41] which is a dataset constituted of 5000 frames taken from a video of a person engaged in a conversation that lasted for 200 seconds. The two blinking detection approaches have been applied to the dataset (native and proposed) and we noticed that our approach detects more than the native one because of single eye management. This is illustrated in

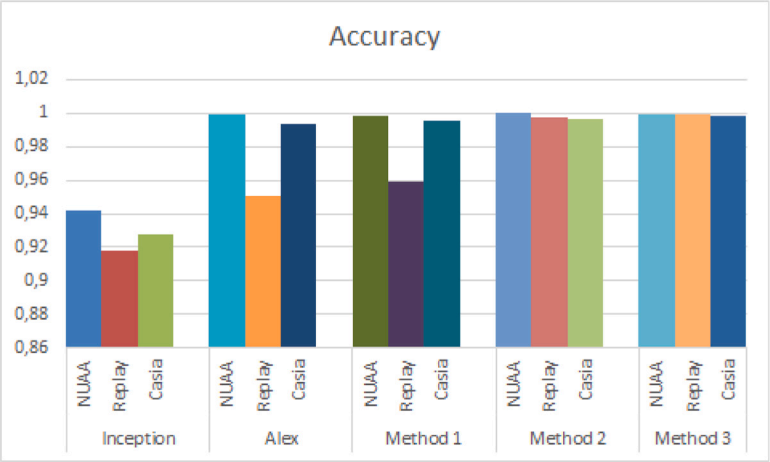

Fig. 10. Accuracy of different approaches tested on NUAA, Replay and Casia dataset.

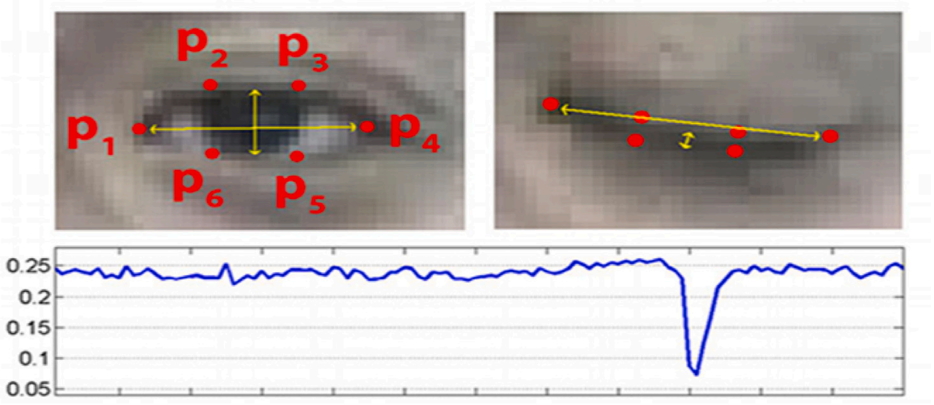

Fig. 11. Detection of blinked eye using landmarks [34].

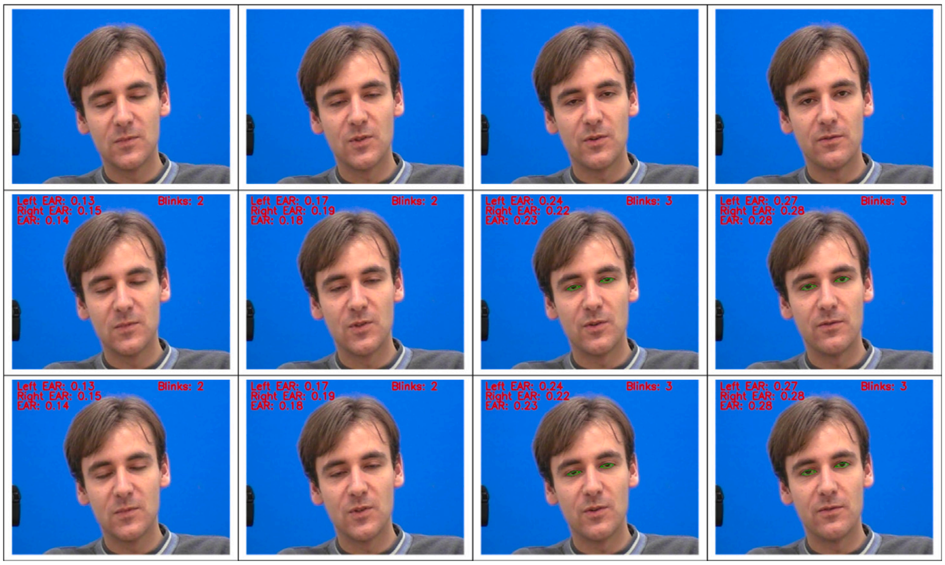

Fig. 12. Detection of blinks on an input images. Line 1: Input frames number 276, 277, 278, 279, ; Line 2: Blink detection using native approach; Line 3: Blink detection using proposed approach.

Table 8 where the proposed approach has a percentage of detection of 77.04% compared to the native approach having a percentage of 68.85%.

5. Conclusion

This paper presented a Facial Signature for enhancing the Identification, Verification and Authentication of Persons. For that, we

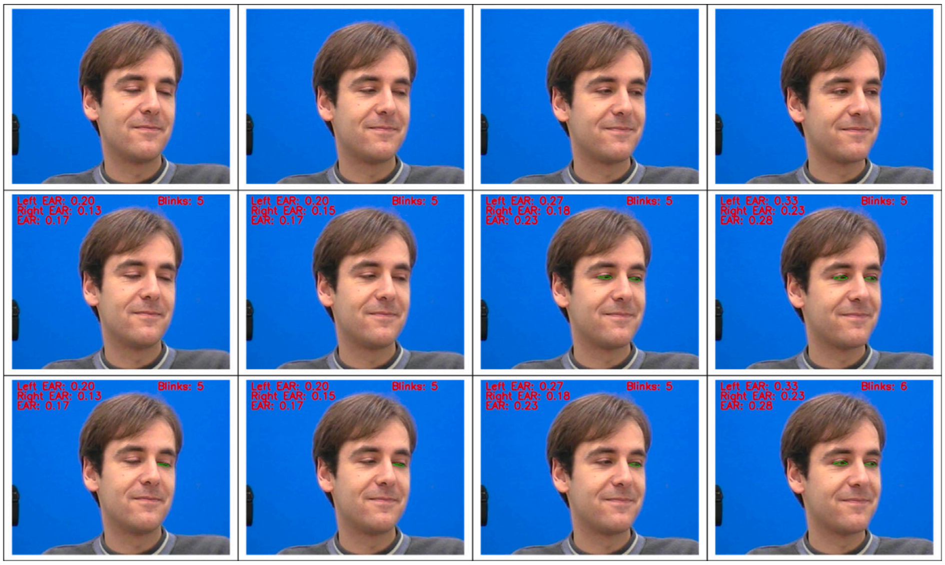

Fig. 13. Detection of blinks on an input images. Line 1: Input frames number 1127, 1128, 1129, 1130, ; Line 2: Blink detection using native approach; Line 3: Blink detection using proposed approach.

Table 8
Number of blinks detected on a Talking face data-set [41].

Method	Time of the video	Number of frames	Number of blinks detected	Percentage
Native blinking method	200 seconds	5000 frames	42/61	68.85%
Proposed blinking method	200 seconds	5000 frames	47/61	77.04%

studied related work on face representation. Later, the FaSIVA proposed method was detailed, where each of its parts was clearly illustrated. This considers three parameters namely the quality of the image, the pattern extracted from the image which is the combination of the pattern extracted from two popular models: the Resnet-50 and Facenet, and finally, a proposed authentication mechanism that takes into consideration two metrics; eye blinking and spoofing verification. A section dedicated to the implementation of FaSIVA and all the steps we went through to train our models which were followed by a testing phase were presented. It was proven that the proposed signature is valid, efficient and robust. This is because it no more focuses only on recognition but also handles operations that enhance image quality as well as authentication mechanisms to reinforce the recognition process and prevent spoofing attacks. From the results obtained, we are able to improve face recognition pipeline by enhancing image quality in case of low resolution images and prevent spoofing attacks of the system using liveness detection. For future work, we intend to investigate more on blinking detection, to accurately detect blinks even in non-frontal faces and in occluded faces. Now that the signature is well defined we are working inserting the ability of detecting adversarial attacks.

CRediT authorship contribution statement

Elie Tagne Fute: Conceptualization, Methodology, Writing – original draft, Formal analysis, Writing – review & editing. **Lionel Landry Sop Deffo:** Methodology, Data curation, Formal analysis, Writing – original draft. **Emmanuel Tonye:** Conceptualization, Writing – review & editing.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work.

References

[1] Kouzani A, He F, Sammut K. Fractal face representation and recognition. IEEE Int Conf Syst Man Cybern 1997;2. http://dx.doi.org/10.1109/ICSMC.1997.638231.

[2] Liu Y, Singleton A, Arribas-Bel D. A principal component analysis (PCA)-based framework for automated variable selection in geodemographic classification. Geo-Spatial Inf Sci 2019;22(4):251–64. http://dx.doi.org/10.1080/10095020.2019.1621549.

[3] Ahonen T, Hadid A, Ainen MP. Face recognition with local binary patterns. In: 8^{th} European conference on computer vision, vol. 3021. 2004, p. 469–91. http://dx.doi.org/10.1007/978-3-540-24670-1_36.

[4] Turk M, Pentland A. Face recognition using eigen-faces. In: IEEE conference on computer vision and pattern recognition, vol. 3. 1991, p. 586–91. http://dx.doi.org/10.1109/CVPR.1991.139758.

[5] Mackiewicz A, Ratajczak W. Principal components analysis (PCA). Comput Geosci 1993;19(3):303–42. http://dx.doi.org/10.1016/0098-3004(93)90090-R.

[6] Sahoolizadeh H, Aliyari Y. Face recognition using eigen-faces, Fisher-faces and neural networks. In: 7^{th} IEEE international conference on cybernetic intelligent systems. 2008, http://dx.doi.org/10.1109/UKRICIS.2008.4798953.

[7] Shashoa NAA, Ahmed N, Jleta I, Abusaeeda O. Classification depend on linear discriminant analysis using desired outputs. In: 17^{th} international conference on sciences and techniques of automatic control and computer engineering. 2016, http://dx.doi.org/10.1109/STA.2016.7952041.

[8] Geng C, Jiang X. Face recognition using sift features. In: 16th IEEE international conference on image processing. 2009, http://dx.doi.org/10.1109/ICIP.2009.5413956.

[9] Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering. In: IEEE conference on computer vision and pattern recognition. 6, (7):2015, p. 815–23. http://dx.doi.org/10.1109/cvpr.2015.7298682.

[10] Liu S, Deng W. Very deep convolutional neural network based image classification using small training sample size. In: 3^{rd} Asian conference on pattern recognition. ACPR, 2015, http://dx.doi.org/10.1109/ACPR.2015.7486599.

[11] Liu1 B-D, Meng J, Xie W-Y, Shao S, Li Y, Wang Y. Weighted spatial pyramid matching collaborative representation for remote-sensing-image scene classification. Remote Sens 2019. http://dx.doi.org/10.3390/rs11050518.

[12] Cao Q, Shen L, Xie W, Parkhi OM, Zisserman A. VGGFace2: A dataset for recognising faces across pose and age. In: IEEE conference on automatic face and gesture recognition. 2018, Cs.CV.

[13] Szegedy C, Vanhoucke V, Ioffe S, Shlens J. Deep residual learning for image recognition. Comput Vis Pattern Recognit 2016. http://dx.doi.org/10.1109/CVPR.2016.308.

[14] Sun J, Cai X, Sun F, Zhang J. Scene image classification method based on Alex-Net model. In: 3rd international conference on informative and cybernetics for computational social systems. 3, 2016, p. 363–7. http://dx.doi.org/10.1109/ICCSS.2016.7586482.

[15] Ilyas BR, Beladgham M, Merit K, taleb Ahmed A. Illumination-robust face recognition based on deep convolutional neural networks architectures. Indonesian J Electr Eng Comput Sci 2020;18(2):1015–27. http://dx.doi.org/10.11591/ijeecs.v18.i2.pp1015-1027.

[16] Sun Y, Liang D, Wang X, Tang X. Deepid3: Face recognition with very deep neural networks. 2015, arXiv:1502.00873 [Cs.CV].

[17] Zhou E, Cao Z, Yin Q. Naive-deep face recognition: Touching the limit of LFW benchmark or not? 2015, arXiv:1501.04690v1 [Cs.CV].

[18] Sun Y, Wang X, Tang X. Deeply learned face representations are sparse, selective, and robust. 2014, arXiv:1412.1265 [Cs.CV].

[19] Zhu Z, Luo P, Wang X, Tang X. Recover canonical-view faces in the wild with deep neural networks. 2014, arXiv:1404.3543 [Cs.CV].

[20] Cao X, Wipf D, Wen F, Duan G, Sun J. A practical transfer learning algorithm for face verification. In: IEEE international conference on computer vision. 2013, http://dx.doi.org/10.1109/ICCV.2013.398.

[21] Chen D, Cao X, Wang L, Wen F, Sun J. Bayesian face revisited: A joint formulation. In: Computer Vision-ECCV. Springer; 2012, http://dx.doi.org/10.1007/978-3-642-33712-3_41.

[22] Taigman Y, Yang M, Ranzato M, Wolf L. Deepface: Closing the gap to human-level performance in face verification. In: Conference on Computer Vision and Pattern Recognition. 2014, http://dx.doi.org/10.1109/CVPR.2014.220.

[23] Islam A, Kowal M, Jia S, Derpanis KG, Bruce NDB. Position, padding and predictions:A deeper look at position information in CNNs. 2021, arXiv:2101.12322v1.

[24] Ren K, Zheng T, Qin Z, Liu X. Adversarial attacks and defenses in deep learning. Engineering 2020;6(3):346–60. http://dx.doi.org/10.1016/j.eng.2019.12.012, Elsevier.

[25] Papernot N, McDaniel P. Extending defensive distillation. 2017, arXiv:1705.05264 [Cs.LG].

[26] Chen B, Ren Z, Yu C, Hussain I, Liu J. Adversarial examples for CNN-based malware detectors. IEEE Acces 2019;7. http://dx.doi.org/10.1109/ACCESS.2019.2913439.

[27] Weiler M, Forre P, Verlinde E, Weilling M. Coordinate independent convolutional network: Isometry and gauge equivariant convolutions on Riemannian manifolds. 2016, arXiv:2106.06020, 7.

[28] Broy M, Wirsing M, Pair C. A systematic study of models of abstract data types. Theoret Comput Sci 1984;33(2):139–74. http://dx.doi.org/10.1016/0304-3975(84)90086-0.

[29] Kamin S. Final data type specifications: A new data type specification method. In: 7th ACM SIGPLAN-SIGACT symposium on principles of programming languages, vol. 28. 1980, p. 131–8. http://dx.doi.org/10.1145/567446.567459.

[30] Wirsing M. Structured algebraic specifications. Theoret Comput Sci 1986;42:123–249. http://dx.doi.org/10.1016/0304-3975(86)90051-4.

[31] Zhang K, Zhang Z, Li Z, Qiao Y. Joint face detection and alignment using multi-task cascaded convolutional networks. IEEE Signal Process Lett 2016. http://dx.doi.org/10.1109/LSP.2016.2603342.

[32] Guo J, Deng J, Xue N, Zafeiriou S. Stacked dense U-nets with dual transformers for robust face alignment. 2018, arXiv:Cs.CV1812.01936.

[33] Jiankang D, Jia G, Xue N, Stefanos Z. Arcface: Additive angular margin loss for deep face recognition. In: Conference on computer vision and pattern recognition. 2019, http://dx.doi.org/10.1109/CVPR.2019.00482.

[34] Soukupová T, Čech J. Real-Time Eye Blink Detection using Facial Landmark. In: 21th computer vision winter workshop. 2016,

[35] Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering. In: Computer vision and pattern recognition IEEE conference. 2015, http://dx.doi.org/10.1109/cvpr.2015.7298682.

[36] Sulayman N. Calculation of accuracy when designing measuring instruments. Meas Technol 2000;43(12):1031–4. http://dx.doi.org/10.1023/A:1010931516493.

[37] Fute ET, Deffo LLS, Tonye E. Eff-vibe: An efficient and improved background subtraction approach based on vibe. Int J Image Graph Signal Process (IJIGSP) 2019;11(2):1–14. http://dx.doi.org/10.5815/ijigsp.2019.02.01.

[38] Deffo LLS, Fute ET, Tonye E. Cnnsfr: A convolutional neural network system for face detection and recognition. Int J Adv Comput Sci Appl (IJACSA) 2018;9(12). http://dx.doi.org/10.14569/IJACSA.2018.091235.

[39] Ito K, Okano T, Aok T. Recent advances in biometric security:A case study of liveness detection in face recognition. In: Asia-Pacific signal and information processing association annual summit and conference. 12, (15):2017, p. 220–7. http://dx.doi.org/10.1109/APSIPA.2017.8282031.

[40] Koshy R, Mahmood A. Optimizing deep CNN architectures for FaceLiveness detection. Entropy 2019;21(423). http://dx.doi.org/10.3390/e21040423.

[41] Zhou H, Liu Y, Liu Z, Luo P, Wang X. Talking face generation by adversarially disentangled audio-visual representation. 2019, arXiv:1807.07860v2 [Cs.CV].