Face anti-spoofing using trainable LBP and margin BCE loss with person identities based metric distance

Meysam Shahbazi, Member, IEEE, Mohammad Ali akhaee, Fellow, OSA,

Abstract—An automated authentication method that makes use of the user's face is one option. Because of substantial advancements in face recognition technology, facial recognition has become increasingly common. Face authentication is not totally safe, however, and an attacker can authenticate by printing the target person's face or replaying a video of him / her instead of the target person, which is a known vulnerability. Academic and industrial research have therefore developed methods and algorithms in this field to increase the security of face authentication systems, which have been tested and proven to work. The goal of this investigation is to determine the difference between the real face image and the phony face image supplied by the attacker. Deep learning algorithms have been used to classify the real image against the fake images provided by the attacker as a result of the increased use of deep learning methods in machine vision problems. Deep learning algorithms have been used to classify the real image against the fake images provided by the attacker. In this dissertation, a novel operator is presented to replace one of the convolution layers in a machine vision system by integrating the classical way of machine vision with deep learning methods. Additionally, in order to improve the classification accuracy between the two categories of real and counterfeit images, a cost function for binary classification with a margin has been proposed, which adds a margin to the samples of the two classes in order to space the samples of the two classes apart. In addition, in order to improve the network's scalability, a specific metric cost function for the problem of face fraud detection has been presented, which makes use of the identities of persons to do this. Furthermore, on certain well-known datasets in this sector, the results are presented, and the overall performance of the suggested approach is reviewed, as well as the execution speed of the algorithm under consideration.

Index Terms—Face authentication, Security of authentication systems, Combining machine vision techniques with deep learning, cost function

I. Introduction

ASSUME that the user has to stand in front of a camera to have his face verified by a face authentication system. Say that an unauthorized user makes a paper copy of a previously approved user's portrait and holds it up to the system camera in order to get their picture taken. To acquire access to another person's personal information, an unauthorized user only needs a photocopy of the authorized user's photo to identify himself in the system.

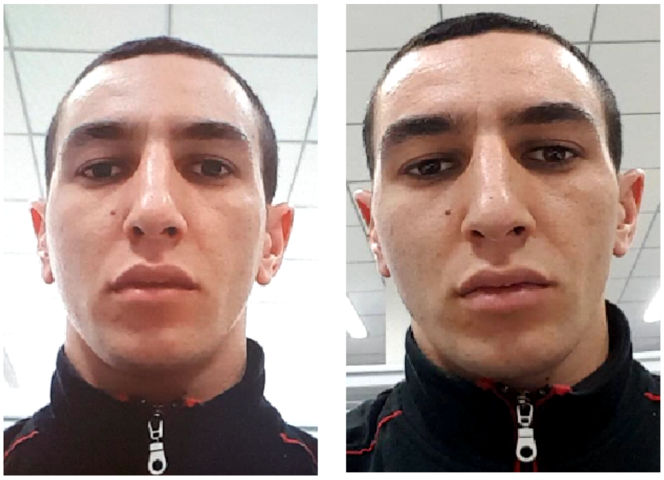

Fig. 1: real and fake example[1]

This is a simple illustration of a security issue with facial authentication systems. When a system contains sensitive and valuable information, security is a top priority.

The primary objective of biometrics is to identify people automatically based on their behavior or biological traits. Face, iris, fingerprints, voice, and gait, for example, are all characteristics that distinguish one person from another. One of the most important biometric characteristics is face recognition. Machine vision-based face detection methods have been around for a long time, and deep learning solutions have recently made face recognition more accurate and ubiquitous. A person's face, on the other hand, is a more familiar representation of them than their fingerprints or sounds.

The growing prevalence of facial recognition in authentication systems raises security concerns. Using social media or remote photography, an adversary can simply and affordably obtain a face image of a target, and then take the necessary procedures to launch an attack. This kind of attack can be carried out using a wide range of instruments. For instance, an attacker could print a paper image of the victim or use a movie or image stored on a digital display. He can also use makeup or a mask to resemble the target's appearance. Among the types of attacks mentioned, the use of image printing and the use of digital displays are more common. Masks are not extensively utilized due to their exorbitant cost and

M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (see http://www.michaelshell.org/contact.html).

J. Doe and J. Doe are with Anonymous University.

Manuscript received April 19, 2005; revised August 26, 2015.

difficult usage.

Given the significance of the issue and security issues regarding face recognition-based authentication systems, academia and industry have conducted numerous studies to address this challenge.

Despite more than a decade of research in this field, face anti-spoofing remains a challenge. One of the obstacles is the attacker's innovative in launching a new attack in a manner that did not previously appear in the network's training data. Another impediment is the disparity in the quality and resolution of attack tools, such as screens and printing paper. This problem becomes more complicated when even humans are unable to distinguish between a real and a fake face. For example, one of the images in figure 1 is a forgery and the other is genuine.

to tackle with this problem

To combat face spoofing, traditional methods [2] and the recently proposed deep learning strategy have been utilized [3]. Deep learning techniques began with simple convolutional neural networks (CNNs) with binary loss [4], however it may reveal arbitrary features and unfaithful spoofing cues, resulting in a lack of generalization. Consequently, several studies benefit from auxiliary supervision[5]. in face anti-spoofing The most well-known auxiliary signal is depth [6], [7], [8], [5], [9], [10]. However, depth supervision demands the synthesis of a depth map for each training sample, which increases both the cost of computing and the complexity of the model. There are also a number of works that leverage temporal information from the sequence of video frames[11], [12], [13], [14]; however, this results in a longer response time, which is undesirable for mobile device and real-time applications.

A. XXXXXXXXXXXXXXXXXXXXXXX

This paper examines strategies for anti-spoofing that use just a color image as input. This method will result in a decrease in system expenses and an improvement in system usefulness. Additionally, only paper printing and video playback are studied, out of the several attack kinds available. While additional assaults, such as the use of a 3D mask, are possible, they are more expensive and harder to perform. As a result, the paper concentrates on more prevalent and executable assaults.

This work blends traditional machine vision techniques with novel deep structural learning techniques to achieve more accurate classification. This structure contains a new operator inspired by the conventional LBP operator, except that, like the convolution operator in deep networks, this operator includes a parameter for learning the best operator for the input data.

Additionally, two additional cost functions are included to improve the network's accuracy and generalization on unseen test data. It is shown in this paper that the suggested strategy, which incorporates a micro texture analysis operator and a novel cost function, improves the classification accuracy and generalization of the system. Additionally, Python is used for implementation, and implementation concerns, and associated issues are described and understood. Additionally, an approach is suggested that utilizes video data to accelerate the network training process. The program's source code is available as open source through a GitHub [1] repository. The program is constructed in such a manner that the outcomes are reproducible.

II. related works

Photometric analysis uses picture texture patterns. Inspecting the magnifying glass scale the most common operator for this kind of analysis is the Local Binary Patterns (LBP) operator. The frequency analysis method uses Gaussian differential filters and cosine conversion. The head, mouth, and eyes move dynamically to engage the facial muscles. In this case, optical flow techniques are generally used. Face separation from the background and frequency information mobility are also utilized. Texture changes between frames are also used. Microtext analysis is one of the software tools addressed in this paper. Using a magnifying lens to examine the texture of facial features may help distinguish between real and fake photos. The grainy texture of the paper contrasts with the natural texture of the human face at this scale. The actual face will also have a different pixel texture than the computer screen image. In terms of light reflection and shadow creation, the real face is not similar to the image printed or shown on a computer. False photographs are also often fuzzy. As a consequence, fraud detection is similar to image quality analysis and cryptography. [2] For the first time, the local binary pattern operator (LBP) is used in face recognition. This operator gives a strong texture description inspired by neighborhood-scale texture. To make judgments in three-dimensional space, Pereira et al. [42] used texture information in the spatial domain and between frames. 2- Deep learning-based methods The LBP operator chooses features. To intelligently choose a feature, deep learning algorithms have been used. Ying et al. [4] pioneered deep learning in facial recognition fraud. This study's strategy is to first identify the face, then enlarge the window chosen for the face to include the face's background. Because background knowledge may help identify fraud. The images are then submitted to an ALEXNET network [41], which uses convolution to extract features and SVM to classify them. Although this was achieved in 2014, relying just on deep neural networks to achieve the required accuracy is inadequate. As a consequence, various ideas for enhancing performance and classification accuracy have been made. This method uses a frame. Three-dimensional convolution is suggested to use information from many frames [43, 44]. Rather than using a convolutional network, [46] and [25] used an LSTM structure [45]. A blend of convolutional layer and manual features may help increase the neural network's accuracy. (2) [9] shows the several modes that may be used to build the structure. This technique may be used in many ways: first extract the manual attribute, then feed it

[1] https://github.com/meysamshahbazi/fas

to a deep network. You may either extract deep features first, then apply manual feature extraction on them, or you can blend deep and manual features and feed them to the classifier. For example, Feng et al. [5] recommend a pre-trained network. The VGG-face network [39] is used, which has been trained to recognize faces in massive volumes of data and fine-tuned for fraud detection. A better grid is created by stacking the values of the grid's middle layers, averaging them, and then decreasing them using the PCA approach. The lower dimension matrices are then input into an SVM classifier. Their approach was to first train a VGG-face neural network using fraud detection data before applying the LBP operator to various network layers. In order to classify it, they used SVM [10]. Continuation of the neural network by Rahman et al. on the LBP operator's input image. The notion of supplemental monitoring [20] was born out of researchers' search for well-designed features. For the first time in this field, Atom et al. [6] used depth as an additional signal. In addition to the supplementary depth signal, Liu et al. [20] used the rPPG signal estimate across successive frames as a face life signal The real depth label for the live face and the zero depth label for the fake face are determined first [6]. [27] Wong et al. [28] used optical flow on neural network properties to estimate depth. For convolutional networks, Yu et al. [8] designed a new structure that puts more emphasis on the central pixel and gives it a different weight than in conventional convolution. There are other methods for calculating the ideal network, such as [8, 18], [29]. The 3DPC-NET structure was designed to use superpoints in three-dimensional space as an auxiliary signal rather than estimating the depth of a plane in two dimensions. Face recognition is a kind of drug detection, according to Yu et al. Face skin differs from printed paper and screen in its composition. That is, it employed the bilateral filter on the deep network's properties to identify the material composition. It is well known that the extra depth signal is expensive and requires further processing to identify depth. It is possible to locate well-constructed features without using depth [26] as George and Marcel demonstrated. A 14 x 14 page is aligned using the DENSNET network [50]. And that instead of a single digit, the real label is a two-dimensional matrix of full length zero or one, with the cost function of binary cross-entropy replacing a neuron on a two-dimensional plane. For the first time, Jurablo et al. [16] used GAN to model and recognize noise in fake photographs. The chance of detecting a fake image rises when the noise associated with fraud detection is quantized

In order to estimate fraud patterns in many picture dimensions, Liu et al. [17] proposed a GAN-based framework. Using the triple cost function also increases scalability. [33] To go along with GAN, Jia et al. Affecting the distance between true and fake samples in diverse datasets. To generate fraudulent patterns, Feng et al. [14] employed a U-Net structure [52] with a triple cost function. To an auxiliary classification network. E.g., Percabu et al. Real samples are around a center, whereas fraudulent samples are one edge away from it. The Real Sample Center updates throughout network training. A cost function was also introduced in the VGG-face network training by You et al. [15] to limit network overfitting and over recognition. Regardless of the label, this function reduces the distance between two sets of data sample sets. To estimate the input image and LBP structure's depth, Zheng et al. [53] used LBP estimation. True photographs have an LBP of 0, whereas fake photos have an LBP of 1.

III. proposed method

This section discusses the suggested method's theoretical roots. The suggested solution entails constructing a trainable operator with an LBP-like formulation and inserting it into the first layer of a traditional convolution network.

This layer is based on LBP since it is critical for identifying spoof patterns rather than concentrating on physical characteristics such as corners, edges, and so on. To begin, we shall express the trainable LBP. The network topology will next be detailed, followed by an explanation of the newly incorporated cost function. To define the trainable LBP operator, a broad definition of the convolution operator and convolutional networks will be provided, as well as an explanation of the logic behind its use in machine vision problems.

The mathematical relationship between the convolution operator and the LBP operator is then shown, and the trainable LBP operator is generated by replicating these two amplifiers.

Two cost functions will be presented in the following sections to maximize the network's accuracy and scalability. The purpose of the first cost function is to distinguish the two classes by a margin, whereas the goal of the second cost function is to push the network to focus on fraudulent qualities rather than physical attributes of the people.

A. Convolution

The convolution operator is a critical component of deep learning networks. This operator contains a core of coefficients that are multiplied by torsion in the input picture, and then an output image is created by sliding over the whole input image.

Applying the convolution operator to the signal is comparable to multiplying the operator's Fourier transform by the input picture's Fourier transform; by multiplying, certain frequencies in the input image may be boosted or diminished, thus filtering the image. By varying the weights applied to the filter core, it is possible to generate images with precise specifications.

For instance, by specifying appropriate weights in the operator, a low-pass filter can be created, and by applying the convolution operator to this low-pass filter, a picture with deleted high frequencies may be formed. Creating multiple filters may be used for a variety of reasons, such as locating an image's edge or reducing picture noise. However, each target needs a custom-designed filter.

The concept behind CNN neural networks is to take filter weights into account and to update the filter coefficients throughout the cost function optimization process, as well as to filter through filters. Available.

Each time a layer of convolution is applied to an image, new characteristics are acquired, and by applying consecutive parameters of parameterized convolution, further conceptual properties are achieved. If optimized with sufficient data, this layered structure may extract semantic features from the picture. This ability to derive meaning from images has resulted in a variety of applications, including categorization, object identification, and face recognition.

B. Trainable LBP

When it comes to identifying spoof in face recognition, it is more crucial to look for characteristics in the picture that indicate whether the face is genuine or fake than it is to evaluate the image's semantic elements. Indeed, the objective is to create a network capable of detecting signals of picture fraud. One of the features of picture fraud is the appearance of the image on a tiny size, which seems difficult to identify at first look. Another defining feature of picture fraud is its prevalence in the majority of the face. To do this, the first stage introduces an operator whose aim is to evaluate the picture texture and use the neural network concept to choose the optimal operator based on the network input data.

The convolution operator and the input image have a relationship in the form of a relation (3.1)

$$CNN = \sigma(\sum_{p \in N} I_p W_p) \tag{1}$$

Where I p denotes the image's brightness in pixels p inside a neighborhood or window with filter dimensions. And W p is the filter's weight in the operator window's p coordinates. Additionally, the function (.) is a nonlinear function.

Additionally, the LBP micro-texture operator has the following relationship (3.2) [2]:

$$LBP = \sum_{p \in N} \sigma(I_p - I_c) 2^p \tag{2}$$

$$\sigma(x) = \begin{cases} 1 & x \geq 0\,; \\ 0 & \text{otherwise}\,. \end{cases}$$

Where is the pixel brightness in the operator window's center. Indeed, in this neighborhood, the value of each pixel is subtracted from the neighborhood's center pixel and a weight of is determined depending on whether the result is larger than or less than zero. This weight is calculated statically based on the contractor's specification. To implement the notion of obtaining optimum weights via data analysis in this amplifier, the definition of this operator must achieve a parametric rather than a station definition. To do this, the weight is altered as shown in Equation (3.3)

$$2^p = e^{p \ln 2} = e^{w_p} \tag{3}$$

Where the is a parameter that changes during optimization to reach the best value for classification. By inserting this parameter in the classical LBP relation, the trainable LBP operator will be obtained in the form of relation (3.4).

$$LBP_{tr} = \sum_{p \in N} \sigma(I_p - I_c) e^{W_p} \tag{4}$$

$$\begin{aligned} \frac{\partial LBP_{tr}}{\partial W_{p^*}} &= \frac{\partial(\sum_{p \in N} \sigma(I_p - I_c) e^{W_p})}{\partial W_{p^*}} \\ &= \sum_{p \in N} \frac{\partial(\sigma(I_p - I_c) e^{W_p})}{\partial W_{p^*}} \\ &= \frac{\partial(\sigma(I_{p^*} - I_c) e^{W_{p^*}})}{\partial W_{p^*}} = \sigma(I_{p^*} - I_c) e^{W_{p^*}} \end{aligned}$$

$$LBP_{tr} = \sum_{p \in N} \sigma(I_p - I_c) W_p \tag{5}$$

$$\frac{\partial LBP_{tr}}{\partial W_{p^*}} = \sigma(I_{p^*} - I_c) \tag{6}$$

In comparison to the convolution operator, this operator will have a more limited perspective of the picture texture. Because convolution multiplies all surrounding pixels by the filter weights and then adds them in a nonlinear function, all neighboring pixels have an effect equal to their corresponding weight in the output. However, since the LBP operator applies a nonlinear function to the difference between each pixel and the center pixel, it examines the picture in more depth and extracts the image's textural properties.

is a nonlinear function that performs a similar purpose to the activation function in neural networks. This function's purpose is to generate nonlinear relationships for the operator, and a critical distinction between the teachable LBP operator and convolution is that the nonlinear function is applied within the resulting operator, whereas in neural network convolutions, the nonlinear function is applied outside the sum operator. Although the Heaviside function is the traditional definition for the LBP amplifier, other nonlinear functions such as Relu and Sign may also be employed.

C. Network structure

Since the LBPtr operator performs micro-scale image processing, it is employed as the initial layer of the deep network. As a result, the network structure will take the following form: (3-1).

The LBPtr operator receives the input picture in the form of three-color channels and outputs it to a network of convolutional layers. The EfficientNet B0 network [37] was employed in this study. And its output will be a flat vector that must be normalized using the output's cost function.

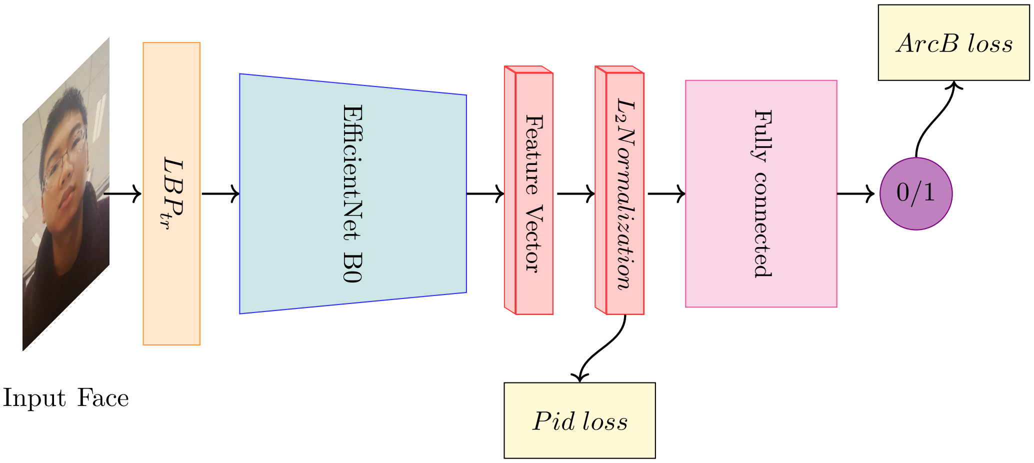

Fig. 2: network

This normalized output will end in another linear layer of a neuron.

The last layer of single neurons will have a value between 0 and 1. And will be categorized according to their values and the threshold level selected for the two classes. The binary cross entropy (BCE) function is the most often used cost function in neural networks for identifying two groups. However, prior research into fraud detection has shown that this cost function alone will fail to identify fraud. To accomplish this, a new classification cost function is created that provides a secure margin for classification, hence increasing network scalability.

D. ARCB loss

When the output of a multi-class classifier (more than two) is produced, the SoftMax activation function is used in the final layer, whereas the Sigmoid activation function is used for two-class classification. Deng et al. In the area of face recognition, which is a multi-class classification, have shifted the cross-entropy cost function (CE) into the cosine space and added a margin to the cost function there [19]. the BCE cost function is revised with the goal of adding a margin to the cosine space.

Assume that the output of the output network is a property of an vector. In conventional decision-making, this vector of dimension d will enter a layer of a neural network with d neurons as inputs and one neuron as output. Finally, the Sigmoid function will be used to convert the output value to a value in the range of one to zero. In decision-making between two classes, the relationship is defined as a function of the binary cross-entry cost.

$$L_{BCE} = -y_i \log P(y_i) - (1 - y_i) \log (1 - P(y_i)) \tag{7}$$

Where is the appropriate label for the feature vector . And is the value of the last layer of neurons; this value is of the probability type, i.e. it has a value between 0 and 1, and the closer it is to 1, the more probable it is to determine which classification output to use. A relationship exists between the output neurons and the X i property vector (3.6).

$$P(y_i) = sigmoid(W^T X_i + b) \tag{8}$$

Where the is the last layer's weight and b denotes the bias value.For the sake of simplicity, the bias is set at zero. Additionally, before to the last layer of the neural network, it normalizes the value of the W weights, normalizes the property, and then scales it to s. This scaling is used to maintain the optimization process's stability. By normalizing the result of the internal multiplication between the weight and the vector, the angle between these two vectors will have the same cosine equivalent attribute.

$$W^T X_i = |W^T||X_i| \cos\theta_i = s \cos\theta_i \tag{9}$$

Additionally, the sigmoid function is defined as Equation $sigmoid(x) = \frac{1}{1+e^{-x}}$ Equation is now translated as Equation by substituting this value into the BCE cost function (3.9).

$$L_{BCE} = -y_i \log \frac{1}{1 + e^{-s\cos\theta_i}} - (1 - y_i) \log (1 - \frac{1}{1 + e^{-s\cos\theta_i}}) \tag{10}$$

Two circumstances exist depending on the value of the real label, which is either zero or one. The first instance. When the label is set to 1, just the first expression in equation (3.9) is shown. It is ideal to maximize the value within the logarithm in this situation, which is similar to the angle between the feature vector and the weight of the final layer approaching zero. To optimize with a margin, it is added by m.

$$y_i = 1 \rightarrow \theta_i = \theta_i + m \tag{11}$$

In the second instance. The second expression in equation (3.9) arises when the actual label value is zero. To do this, the expression within the logarithm must be maximized, which is equal to getting the angle between the weight and the property vector closer to. To optimize with the margin, the angle between the two vectors is subtracted from a constant value of the margin m.

$$y_i = 0 \rightarrow \theta_i = \theta_i - m \tag{12}$$

By substituting marginal angles in Equations (3.10) and (3.11) in reference to the BCE cost function recast in cosine space (3.9), we get the ARCB cost function as Equation (3.12).

$$L_{ArcB} = -y_i \log \frac{1}{1+e^{-s\cos(\theta_i+m)}} - (1-y_i)\log(1-\frac{1}{1+e^{-s\cos(\theta_i-m)}}) \tag{13}$$

This cost function not only separates the attributes of the two classes, but also adds a margin to the cosine space properties of the two distinct classes. This margin causes the network weights to be altered throughout the optimization phase to boost the network's generalizability. The difference between the classical cost function and the cost function with margin is seen in Figure (3-2).

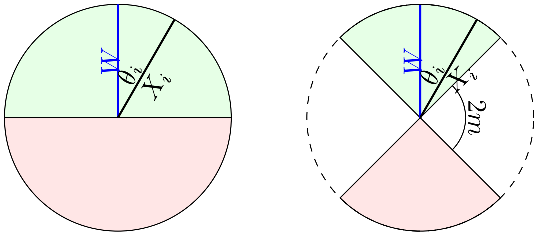

Fig. 3: comparing arcb and bce

f the cost function is optimized properly, it causes the property vectors in the cosine space to be arranged in the final layer in such a manner that the angle between the new sample and the weight vector approaches zero in the case of label one, and is inclined to in the case of label zero. Additionally, the impact of adding margins on the divisibility of the two classes is readily apparent. On the ideal left, Figure (2-3) illustrates the outcome of separating feature vectors using the BCE cost function. And to the right of the ARCB cost function, a margin in the cosine space is used to divide the feature vectors.

E. PID loss

There are multiple live examples and several counterfeit samples for each individual in the extant datasets in the area of face anti spoofing. That is, a person's face has been utilized to gather data and a sample of his live and false video has been captured. There is an identical look characteristic in both the genuine and false video of the person in the database, including the parts of his face that are distinct from the other person. On the other hand, it is preferable for the network to concentrate on the indicators of spoof on the face throughout the training phase, rather than on the facial characteristics. Since the network's primary input picture contains the face and facial characteristics of a person, the network will tend to focus on those aspects for various samples, which is undesirable. As a result, a penalty is applied to the cost function in this section. In this scenario, the network's objective is to disregard physical traits of people in favor of qualities associated with fraud. Assume the following is the output property vector of the attribute extraction step for the k-th individual designated l: $X_k^l \in R^d, I \in \{0,1\}, K \in \{1,2,...,M\}$

This indicates that the database has the sample tag for a number zero or one, as well as the sample tag for a number M of distinct individuals.

Each phase of training assumes the existence of N batch size. There are $\binom{N}{2}$ attribute vector pairings among them, of which two states are significant.

The first instance occurs when two attribute vectors in pairs are associated with the same individual but have distinct labels. That is Given that the physical traits of the individual represented by the primary input picture are identical, it is important to maximize the distance between these two samples in this situation. By increasing this distance, the network is compelled to focus on the feature that distinguishes the two cases, rather than on the physical traits of the individuals, which is the difference in the label of these two feature vectors, one real and one fictitious.

This mode is seen in Figure (3-3). In this illustration, the top image is a real one, while the bottom one is a fake. Due to the similarity of these two photos, the output of their feature vectors may be identical. In the following d space, feature vectors are shown as stars. This distance must be increased as much as possible. Thus, in the first scenario, the network serves as a conduit for communication (3.14).

$$\max_{\Theta} d(X_{k_1}^{l_1}, X_{k_2}^{l_2}) = \min_{\Theta} \max(0, M - d(X_{k_1}^{l_1}, X_{k_2}^{l_2})) \tag{14}$$

Where denotes the set of lattice weights and d is the Euclidean distance between the two property vectors adjusted as Equation (3.15).

$$d(X_1, X_2) = ||\frac{X_1}{||X_1||} - \frac{X_2}{||X_2||}|| \tag{15}$$

Because the cost function must be reduced during optimization, increasing the distance between the two feature vectors is equal to decreasing the value $\max(0, M - d(X_{k_1}^{l_1}, X_{k_2}^{l_2}))$ M is a hyperparameter in this context; if the distance between two attribute vectors is more than this value, the output value is zero; if it is smaller, the distance to this value M is the cost value. Due to the inequality in Equation (3.16), the greatest distance between two property vectors in normalized space is 2, and hence the cost value of M is two in the implementation of this function.

$$||\frac{X_1}{||X_1||} - \frac{X_2}{||X_2||}|| \leq ||\frac{X_1}{||X_1||}|| + ||\frac{X_2}{||X_2||}|| \rightarrow d(X_1, X_2) \leq 2 \tag{16}$$

The second instance is when two attribute vectors in pairs have the same label but are associated with distinct individuals. Mathematically, . Due to the physical qualities of the two properties vectors, they may have a substantial distance in the property space in this situation. It is desired to minimize the distance between the two feature vectors in this scenario. In this situation, the network is required to pick the feature image in such a manner that the distance between the two feature vectors is as minimal as possible, and by accomplishing this aim, the extracted features will be more focused on fraud detection characteristics than on physical traits of persons. have. This circumstance is shown in Figure (3-4). The two input photographs in this example are both fictitious but belong to distinct individuals. The star marks the location of the feature vector corresponding to these two inputs in the subsequent d space in this picture. Due to the fact that two individuals have very distinct physical traits, the distance between their respective property vectors may be quite varied. In this scenario, it is meant to shorten this distance. Thus, in mathematical words, the cost function in this situation will take the shape of a relation (3.17).

$$\min_{\Theta} d(X_{k_1}^{l_1}, X_{k_2}^{l_2}) \tag{17}$$

Finally, the cost function will be dependent on the database's users' IDs as relation (3.18):

$$\begin{aligned} L_{PiD} = \sum_{l_1 \neq l_2, k_1 \neq k_2} & \frac{1}{N_i} d(X_{k_1}^{l}, X_{k_2}^{l}) \\ & + \frac{1}{N_j} \max(0, M - d(X_{k}^{l_1}, X_{k}^{l_2})) \end{aligned} \tag{18}$$

Where is the number of pairs of samples sharing the same tag attribute but belonging to a different individual in the batch, and denotes the number of pairs of samples sharing the same tag attribute but sharing the same ID.

The cost function is formed by selecting all pairings that have the same identifier condition of the same label or the same identifier condition-different label from the N sample in each training step and inserting their Euclidean distance in Equation (3.18). When this cost function is reduced, the network is steered toward identifying useful traits for detecting fraud and away from features associated with people's appearance. Finally, the entire cost of network training will be expressed as a percentage (3.19).

$$L_{overal} = \lambda_1 L_{ArcB} + \lambda_2 L_{PiD} \tag{19}$$

Where and are hyper parameters indicating the degree of focus placed on each.

IV. implementation details

The Python programming language and the Pytorch library are used in this article. This library is a very capable modeler of deep networks. Pytorch's greater versatility compared to other tools makes it simpler to create new functions and unexpected operations.

This paper offers a new LBP operator and a novel cost function that are not readily accessible as modules in deep learning programs but can be implemented using Pytorch computation streams.

To implement a new operator with a learnable argument, a class derived from nn.Module must be written. This will be capable of both forward and reverse computation and can be utilized in deep network computing.

To enable this class to have learning parameters, the class parameter variable must be written using nn.Parameter. If this operator is employed as a layer in a network, the LBP operator parameters will be included in the network parameters, and optimization will result in these parameters being updated.

Each time input is passed to the network after the extraction block, the property of a vector is retrieved, which must be normalized at each step before being used in the two cost functions introduced. One normalization will serve for network testing, since there will be no change in weights. The ARCB function is implemented using Pytorch functions to ensure computational stability. Drop out [36] was employed in the last layer after normalizing the feature vector and before the classifier to avoid overfitting the data. To implement the cost function based on the identities of the individuals, in addition to the input picture and image label, an identifier in the form of a number is required. In current datasets, the ID number may be deduced from the video file's name. The network has been optimized using the Adam method [35].

To load and prepare data, the Pythoch library has predefined methods and classes that will automatically utilize the photographs in a folder, but owing to the video nature of the data and the unique cost function presented, it is not feasible to use pre-defined functions.

In certain datasets, there is a file for face coordinates that can be extracted from each video frame, the portion connected to the face, and instead of utilizing the whole frame as input to the network to be supplied, just the portion of the face with a little amount of the image's backdrop may be used. In datasets that lack this coordinate file, the faces of the frames are located and recorded in a text file using the MTCCN technique [32]. All of the datasets presented here are in the form of video. Since the suggested technique operates on a single image, one of the more practical aspects of teaching with video data is preparing the data for training. One method is to convert the video to a picture and store it on the hard drive. This, however, will occupy a significant amount of disk space and significantly slow down the training process, since the pictures will need to be reloaded from drive to RAM during training.

On the other hand, since the samples in the two classes are not equal, it is important to have nearly the same amount of films from each class in each category in order to stabilize the ARCB cost function. On the other hand, in order for the cost function based on person IDs to operate effectively, the video must be sufficiently dispersed within each category so that there are distinct examples of individuals with distinct IDs and labels inside each category. Additionally, it is vital for the data to be as random as possible in order to provide additional uncertainty for the network during training. To implement the approach described in this paper, the video will first be put into RAM in batches, and then a frame will be randomly picked from each video in each batch. Finally, there will be a frame for training in each batch. Had. The next phases will use the same videos that were put into RAM, and this procedure will continue as long as there is a frame in the films. Then another video genre will be chosen, and training will continue on all videos.

Because the number of frames is repeated and the training stages are repeated sequentially after picking a number of films, and the subsequent frames of a video are visually near to one another, it is important that the data has a greater degree of uncertainty. This is accomplished via the use of random data augmentation techniques. To do this, converters are utilized that randomly rotate each input picture. To avoid over-fitting, the technique of mistakenly removing a portion of the input picture was applied [40]. Additionally, when the face is to be cut off from the backdrop, this is done using a random window; hence, each time the data is loaded, the location of the face in the cropped picture is random and will not always be in the image's center. Figure (4-1) demonstrates how to cut the face randomly with the backdrop. The blue rectangle represents the person's face in this photograph, while the colorful rectangles are randomly picked for each face selection.

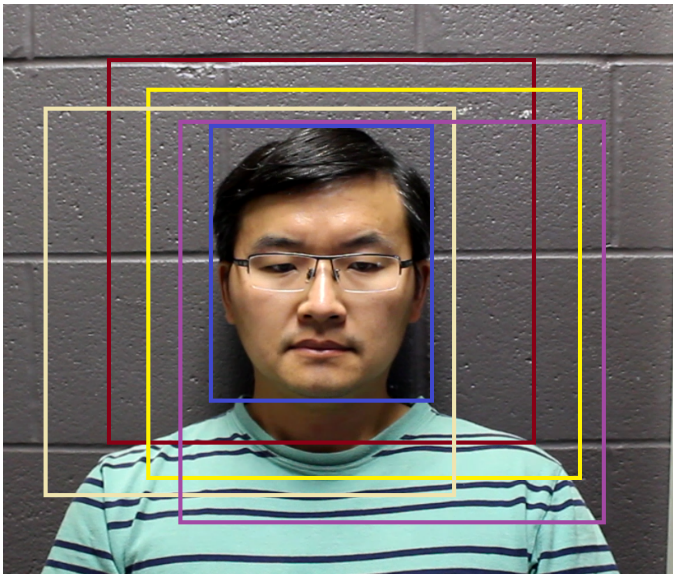

Fig. 4: random cropping around face

To create the data loading class, a custom batch sampler was built, as well as a custom batch sampler function to execute the technique of randomly choosing video and reusing successive video frames. This function is implemented using the Python programming language's iteration notion.

evaluation metrics:

The fraud detection issue is a two-class classification problem that often does not provide an equal number of genuine and counterfeit samples when evaluated. As a result, the network accuracy criteria, defined as the number of properly predicted samples divided by the total number of samples, is insufficient for evaluating network performance.

This is accomplished by using a criterion known as equal error rate and visualizing it for various thresholds in the form of an equal error rate diagram.

Two modes are critical in constructing this graphic. Sample acceptance error rate, which indicates that the tag was really a real person but was expected to be a phony person. Additionally, there is a rejection error rate, which indicates that the sample contains a fictitious label but is projected as a live face.

$$FAR = \frac{number\ of\ false\ accepted\ samples}{total\ number\ of\ fake\ samples} \tag{20}$$

$$FAR = \frac{number\ of\ false\ rejected\ samples}{total\ number\ of\ real\ samples} \tag{21}$$

This value is often determined using one of the factors as a threshold. For example, the value of a single layer of neurons in the last layer of a neural network with a sigmoid activation function will be between zero and one. A judgment is made to forecast the sample label by setting a threshold level and comparing the number of neurons in the final layer to this threshold level. The error rate is equal to the figure obtained by multiplying FAR by FRR.

The graph in Figure (4-2) illustrates this criteria at various threshold values. In datasets with three parts: training, development, and test, the network weights are typically acquired on the training data, whereas the parameter $_{E}ER*$ is obtained on the development data. Additionally, the standard half of the error rate is specified in the test section as Equation (4.5).

$$\tau_{EER} = \arg\min_{\tau} |FAR(\tau) - FRR(\tau)| \tag{22}$$

$$EER = FAR(\tau_{EER}) = FRR(\tau_{EER}) \tag{23}$$

$$HTER = \frac{FAR(\tau_{EER}) + FRR(\tau_{EER})}{2} \tag{24}$$

We may examine the network's performance by evaluating the equal error rate diagram. The less the FRR and FAR curves overlap, the more accurate the network. The FRR and FAR values around the junction also indicate how far apart the two classes are separated by the network.

Another criteria for assessing the usage of ISO / IEC 30107-3 is the definition of attack presentation classification error rate (APCER) and excellent presentation classification error rate (BPCER), the latter of which is similar to the FRR. However, APCER is comparable to the maximum FAR for many assault instruments.

A printed paper assault or a replay attack are both examples of attack tools. Additionally, the average categorization error rate is defined as the product of the APCER and BPCER averages.

$$APCER = \max_{PAI=1,\ldots,C} FAR_{PAI} \tag{25}$$

$$ACER = \frac{APCER + BPCER}{2} \tag{26}$$

V. experimental results

This section evaluates the proposed method's accuracy on a variety of datasets. To begin, the proposed approach is evaluated for efficacy on the Replay database, which is a relatively small database.

This is done in order to proof of concept. Following that, accuracy is provided for more datasets. To assess the offered strategies and their influence on accuracy improvement, an ALEXNET network with a BCE cost function is initially employed. Figure illustrates the curve of the equal error rate in this situation (4-3). As can be observed, using the 0.13 threshold level for the last neuron results in an unnoticed error of 7.3 percent. However, it should be noted that not only the magnitude of the error is significant, but also the performance of the graph in other sections of the threshold level, and at 0.6, the FRR error value is almost 80%, which is very high. Additionally, around the 0.13 threshold level, the error value rises when the threshold level is changed slightly.

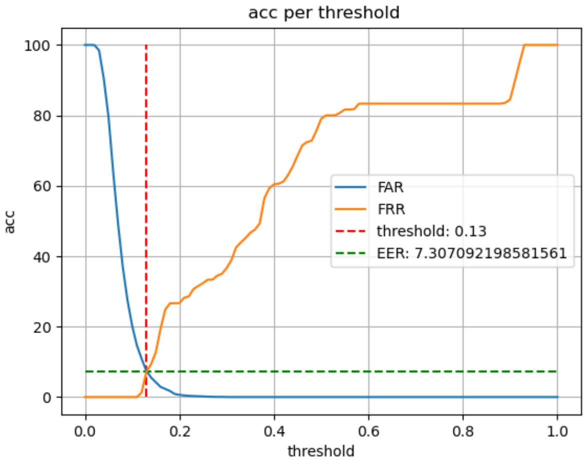

Fig. 5: EER

The diagram in Figure 4-4 is created by training the LBP operator before to ALEXNET with a BCE cost function.

As can be observed, by employing just one LBP layer before to ALEXNET, the error rate was decreased to 0%. Additionally, the error status at the threshold has been enhanced. Given that the inclusion of a layer of trainable LBP operators adds minimal computational complexity to the network, Figure compares the other network training diagram with the BCE cost function and the efficient net B0 network (4-5).

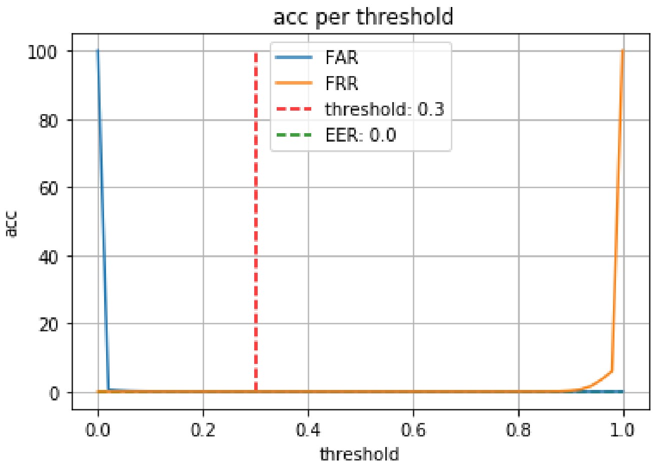

Fig. 6: eer

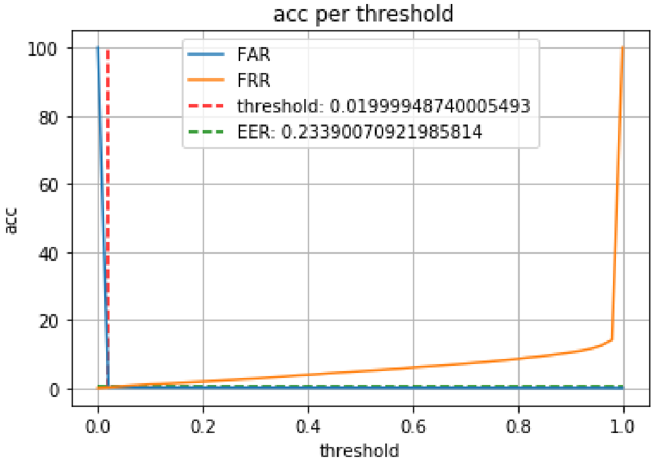

Fig. 7: eer

This figure demonstrates that using a complicated network does not always result in the intended outcome. It is important to note that this graphic does not imply that the LBP layer with ALEXNET is more powerful than the Efficient network. Rather than that, in this specific application and replay datasheet, which contains a little quantity of data, utilizing a simpler but smarter network, depending on the problem, results in increased accuracy.

At the moment, just the ALEXNET network without the LBP operator is employed, but the newly released ARCB cost function is used in place of the BCE function.

The design in Figure (4-6) demonstrates that it is possible to alter the cost function without altering the structure. The graphic is symmetrical in comparison to the preceding ones. The error rate is zero near the threshold level in this figure, but grows as the distance between the threshold level and the values 0 and 1 increases. This margin impact on the ARCB cost function is what results in the separation of two classes by a margin

The basic ALEXNET structure is now employed, and

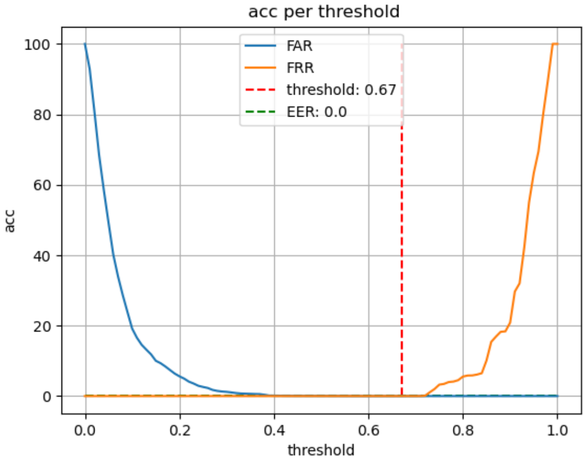

Fig. 8: arcb

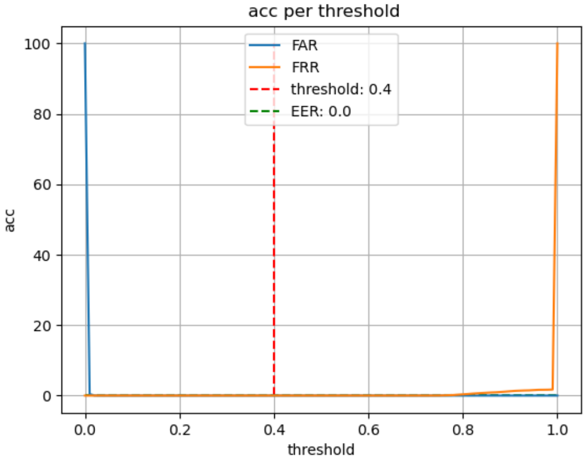

Fig. 9: pid

the classifier's cost function is the BCE function, but a cost function depending on the person's identification has been added as well. This situation is shown graphically in Figure (4-7). As can be observed, the error between the 0 and 0.8 criteria is nil. Which demonstrates that the two classes are sufficiently divided.

Until now, the impact of each suggested approach has been investigated independently. To proceed with the chapter, all approaches are combined. And the extraction network is a characteristic of an Efficient network. Additionally, to aid in the network's convergence, the feature extraction part makes use of weights learned on the image-net database, although these weights vary during the training process. The picture resolution of the MSU and CASIA datasets is greater than that of the Replay datasets. In contrast to replay datasets, which are divided into three components: training, development, and testing, these datasets are divided into two components: training and testing. The value of the equal error rate is presented in the data test section of Table (4-1).

TABLE I: casia and msu

Dataset	EER (%)
CASIA	0.54
MSU	0.0

Given the age of these two datasets, it is not difficult to achieve zero precision. Recent research in this area has focused on accuracy using the SIW and OULU datasets. These two datasets are more recent and greater in size than the preceding two. As a result, the majority of these two datasets have been included into contemporary studies. Each of these two datasets has a unique technique that demonstrates distinct methods for assessing the model's generalization.

The first protocol analyses changes in face state using the SIW database. For training purposes, just the first 60 frames of each video are utilized; however, for testing purposes, all frames from the test movies are used. Because the user does not move his face during the first 60 seconds of each video, the training data comprises only photos of the face in a fixed location in front of the camera. However, the test data covers all of the video's face movement phases. This protocol demonstrates the proposed model's generalization across many face modalities. The outcome of this instance is summarized in Table (4-2) along with a comparison to many well-known methodologies.

TABLE II: siw 1

ACER	BPCER	APCER	Method
3.58	3.58	3.58	[5] Auxiliary
0.25	0.50	0	[15] LGSC
1	-	-	[14] STASN
0.12	0.17	0.07	[7] CDCN
0.4	0.17	0.64	[9] SGTD
0.4	0.17	0.69	[16] 3DPC-NET
0.13	0.12	0.14	ARCB+PID

In the second protocol, one of the four kinds of replay assaults is reserved for testing each time, while the other three are used for network training. Thus, this technique has four distinct modes in which the mean and variance accuracy are presented. This protocol is intended to evaluate the suggested method's performance against an unobserved replay assault. The findings are summarized in Table (4-3).

TABLE III: siw 2

ACER	BPCER	APCER	Method
0.57 ±0.69	0.57 ±0.69	0.57 ±0.69	[5] Auxiliary
0±0	0±0	0±0	[15] LGSC
0.28±0.05	-	-	[14] STASN
0.04±0.5	0±0.09	0±0	[7] CDCN
0.02±0.04	0.04±0.08	0.0±0.0	[9] SGTD
0.45±0.14	0.43±0.06	0.46±0.28	[16] 3DPC-NET
0.0087±0.0151	0.01±0.0173	0.0075±0.0129	ARCB+PID

Additionally, the OULU database contains four distinct protocols, the correctness of which is demonstrated in this paper's first and second protocols. Three separate sites have been used to photograph the OULU database. The first protocol involves training on films linked to the first and second positions, while the second protocol involves doing the test on movies connected to the third place. This protocol is being offered with the intent of analyzing the suggested method's capabilities by shifting the images.

The second protocol stores two printed paper attacks and two replay attacks in the database; one print attack and one replay attack are used for training, while the other print and replay attack is used for testing. The goal of this procedure is to assess instruments that are not often encountered during training. The first and second procedures report on the correctness of the model described in Table (4-4).

TABLE IV: oulu1

ACER	BPCER	APCER	Method
5.7	8.9	2.5	[17]GFA
1.6	1.6	1.6	[5] Auxiliary
1.5	1.7	1.2	[18] FaceDs
0.4	0	0.8	[15] LGSC
1.9	2.5	1.2	[14] STASN
0.2	0	0.4	[7] CDCN
1.0	0.0	2.0	[9] SGTD
0.42	0	0.83	[19] DeepPixBis
1.1	1.3	0.8	[20]STDN
1.2	0	2.3	[16] 3DPC-NET
2.29	2	2.58	ARCB+PID

TABLE V: oulu2

ACER	BPCER	APCER	Method
1.9	1.3	2.5	[17] GFA
2.7	2.7	2.7	[5] Auxiliary
4.3	4.4	4.2	[18] FaceDs
0.7	0.6	0.8	[15] LGSC
2.2	0.3	4.2	[14] STASN
1.3	0.8	1.8	[7] CDCN
1.9	1.3	2.5	[9] SGTD
6.0	0.6	11.4	[19] DeepPixBis
1.9	1.6	2.3	[20] STDN
3.0	2.8	3.1	[16] 3DPC-NET
0.97	0.97	0.97	ARCB+PID

A. corss dataset test

As seen in the preceding sections, achieving an error rate close to zero is not out of reach using new deep learning algorithms. However, how the suggested model performs on unknown data with varying distributions remains a difficult and critical subject in academic study. While a model may achieve great accuracy on a data set with a certain distribution, it performs badly in the actual world.

Thus far, the findings have shown the model's accuracy inside the database. Another critical challenge in the realm of fraud detection is the reproducibility of tests performed on two distinct datasets. This is accomplished by training the model on one dataset and testing it on another.

To validate the model's accuracy while comparing datasets, the network is trained on the CASIA dataset and then evaluated on the Replay dataset. Table (4-5) summarizes the outcome of this case, as well as the accuracy of previous investigations.

TABLE VI: cross test resulat

HTER %	Method
31.5	[14] STASN
17	[9] SGTD
27.6	[5] Auxiliary
28.5	[18] FaceDs
21.4	[17]GFA
27.4	[15] LGSC
23.4	[16] 3DPC-NET
21.25	ARCB+PID

B. computational cost

a metric for measuring the neural network's speed of execution A computer's ability to do floating point operations

Afterwards, when the network has been completed. This value is expressed in (Mac.) units. FLOPs (whatever the value)

The smaller the network, the cheaper the computing cost, and hence the faster the network will be. Comparison of the computing costs of the approaches that were proven accurate in the preceding section is shown in Table 7.4.

The proposed approach of this dissertation is also described, along with the computing costs.

From this table, it is clear that the proposed method's computing costs are a long way off.

Other procedures are employed. EfficientNet B0 is the network's primary processing mechanism.

Comparatively speaking, the network's computing costs will be substantially lower. There is no need to have a large amount of computing power in order to use this strategy.

While minimizing processing costs, it should be emphasized that the suggested method's accuracy is in many circumstances superior to other approaches and in some cases Accurately competitive with current methods.

Unlike [10] CDCN, [8] Auxiliary and [43] SGTD methods and STASN, [8] Auxiliary methods, the proposed method did not use depth as an auxiliary signal. There are no video sequences required for the evaluation of SGTD, and it functions as a single frame.

TABLE VII: flops compariation

FLOPs	Method
50.9 GMac	[5] Auxiliary
47.48 GMac	[7] CDCN
39.4 GMac	[20] STDN
4.64 GMac	[19] DeepPixBis
9.53 GMac	[15] LGSC
15.38 GMac	[17] GFA
1.82 GMac	[16] 3DPC-NET
400.39 MMac	ARCB+PID

VI. Conclusion

The current state of the art in the area of face authentication system security was studied in this paper. Historically, techniques have relied heavily on auxiliary cues such as depth. Consecutive video frames have also been employed in a variety of ways to determine if a face is genuine or not. This paper develops a system based on the use of a single frame. Additionally, the suggested solution eliminates the necessity for a depth signal as an auxiliary signal. However, the suggested technique in the first and second protocols achieved comparable accuracy to existing methods in both big and fresh datasets in this sector. Due to the fact that the suggested approach's primary processing is based on an efficient net network, the computational volume of the proposed technique is minimal. It has a quick reaction time owing to the utilization of a frame. This paper presents a novel CNN-scalable LBP-based operator. Additionally, when the cost function with margin evolved, the network resolution grew. Additionally, the implementation of a cost function based on the individual's ID improved the network's generalizability. The benefit of utilizing the cost function is that it increases accuracy without increasing the network's computing burden. Thus, although the suggested solution takes extra training time, the network test duration remains constant. In this study, an efficient net was utilized. Subsequent study may include the usage of buildings that were initially created. Additionally, it might be beneficial to improve the accuracy of the network's attention structure. Increasing accuracy using a new structure may be accomplished by using a video sequence rather of a frame. To improve texture analysis, the LBP amplifier may be enlarged to fit in all levels of the network rather to only its cannula. Instead of a neuron, write. The cost function that is dependent on the person ID may be replaced by other features such as the attack tool. Additionally, employing depth in conjunction with the suggested approach may improve accuracy. The emphasis of this work is on print and replay assaults. This category contains datasets that involve mask assaults. Further study might be conducted by using a technique similar to the suggested approach to datasets that include RGB and IR pictures.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

References

[1] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid, "Oulu-npu: A mobile face presentation attack database with real-world variations," in 2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017). IEEE, 2017, pp. 612–618.

[2] R. Ramachandra and C. Busch, "Presentation attack detection methods for face recognition systems: A comprehensive survey," ACM Computing Surveys (CSUR), vol. 50, no. 1, pp. 1–37, 2017.

[3] Z. Yu, Y. Qin, X. Li, C. Zhao, Z. Lei, and G. Zhao, "Deep learning for face anti-spoofing: A survey," arXiv preprint arXiv:2106.14948, 2021.

[4] J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," arXiv preprint arXiv:1408.5601, 2014.

[5] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 389–398.

[6] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based cnns," in 2017 IEEE International Joint Conference on Biometrics (IJCB). IEEE, 2017, pp. 319–328.

[7] Z. Yu, C. Zhao, Z. Wang, Y. Qin, Z. Su, X. Li, F. Zhou, and G. Zhao, "Searching central difference convolutional networks for face anti-spoofing," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5295–5305.

[8] R. Shao, X. Lan, J. Li, and P. C. Yuen, "Multi-adversarial discriminative deep domain generalization for face presentation attack detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10 023–10 031.

[9] Z. Wang, Z. Yu, C. Zhao, X. Zhu, Y. Qin, Q. Zhou, F. Zhou, and Z. Lei, "Deep spatial gradient and temporal depth learning for face anti-spoofing," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5042–5051.

[10] Z. Wang, C. Zhao, Y. Qin, Q. Zhou, G. Qi, J. Wan, and Z. Lei, "Exploiting temporal and depth information for multi-frame face anti-spoofing," arXiv preprint arXiv:1811.05118, 2018.

[11] J. Gan, S. Li, Y. Zhai, and C. Liu, "3d convolutional neural network based on face anti-spoofing," in 2017 2nd international conference on multimedia and image processing (ICMIP). IEEE, 2017, pp. 1–5.

[12] H. Li, P. He, S. Wang, A. Rocha, X. Jiang, and A. C. Kot, "Learning generalized deep feature representation for face anti-spoofing," IEEE Transactions on Information Forensics and Security, vol. 13, no. 10, pp. 2639–2652, 2018.

[13] Z. Xu, S. Li, and W. Deng, "Learning temporal features using lstm-cnn architecture for face anti-spoofing," in 2015 3rd IAPR asian conference on pattern recognition (ACPR). IEEE, 2015, pp. 141–145.

[14] X. Yang, W. Luo, L. Bao, Y. Gao, D. Gong, S. Zheng, Z. Li, and W. Liu, "Face anti-spoofing: Model matters, so does data," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3507–3516.

[15] H. Feng, Z. Hong, H. Yue, Y. Chen, K. Wang, J. Han, J. Liu, and E. Ding, "Learning generalized spoof cues for face anti-spoofing," arXiv preprint arXiv:2005.03922, 2020.

[16] X. Li, J. Wan, Y. Jin, A. Liu, G. Guo, and S. Z. Li, "3dpc-net: 3d point cloud network for face anti-spoofing," in 2020 IEEE International Joint Conference on Biometrics (IJCB). IEEE, 2020, pp. 1–8.

[17] X. Tu, Z. Ma, J. Zhao, G. Du, M. Xie, and J. Feng, "Learning generalizable and identity-discriminative representations for face anti-spoofing," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 11, no. 5, pp. 1–19, 2020.

[18] A. Jourabloo, Y. Liu, and X. Liu, "Face de-spoofing: Anti-spoofing via noise modeling," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 290–306.

[19] A. George and S. Marcel, "Deep pixel-wise binary supervision for face presentation attack detection," in 2019 International Conference on Biometrics (ICB). IEEE, 2019, pp. 1–8.

[20] Y. Liu, J. Stehouwer, and X. Liu, “On disentangling spoof trace for generic face anti-spoofing,” in European Conference on Computer Vision. Springer, 2020, pp. 406–422.

PLACE
PHOTO
HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.