International Journal of Communication Networks and Information Security
*ISSN: 2073-607X, 2076-0930*
*Volume 15 Issue 01 Year 2023*

# Restrictive Voting Technique for Faces Spoofing Attack

**Mahmoud Omara**
*Faculty of Computer and information sciences, Ain Shams University, Egypt*
*mahmoud.omara@cis.asu.edu.eg*
**Mahmoud Fayez**
*Faculty of Computer and information sciences, Ain Shams University, Egypt*
*mahmoud.fayez@cis.asu.edu.eg*
**H. Khaled**
*Faculty of Computer and information sciences, Ain Shams University, Egypt*
*heba.khaled@cis.asu.edu.eg*
**Said Ghoniemy**
*Faculty of Computer and information sciences, Ain Shams University, Egypt*
*ghoniemy1@cis.asu.edu.eg*

***Article History***

Received: 18 April 2023
Revised: 17 May 2023
Accepted: 9 June 2023

***Abstract***

Face anti-spoofing has become widely used due to the increasing use of biometric authentication systems that rely on facial recognition. It is a critical issue in biometric authentication systems that aim to prevent unauthorized access. This paper proposes a modified version of majority voting that combines the votes of six classifiers for multiple video chunks in order to enhance the accuracy of face anti-spoofing. The used approach involves sampling sub-videos of 2 seconds each with a one-second overlap and classifying each sub-video using multiple classifiers. Classifications for each sub-video across all classifiers are ensembled to decide the complete video classification. The main focus is on the False Acceptance Rate (FAR) metric to highlight the importance of preventing unauthorized access. The proposed method was assessed using the Replay Attack dataset [1] and yielded a FAR of zero. The Half Total Error Rate (HTER) and Equal Error Rate (EER) were also reported and gained a better result than most state-of-the-art methods. The experimental results demonstrate that the proposed method reduces the FAR to a significant extent, which is critical for real-world face anti-spoofing applications.

***Keywords: Machine-learning, Face Anti-spoofing, SVM, LBP, Voting Techniques, Majority Voting***

## 1. Introduction

Facial recognition technology is widely used in computer-based systems for identifying or verifying the identity of individuals. These systems are commonly employed in various applications, including access control, security, and personal identification. However, most facial recognition systems are susceptible to attacks, specifically face spoofing attacks.

Face spoofing attacks refer to the attempts to bypass or deceive facial recognition systems by presenting a fake or manipulated image of a face for identification or verification. These attacks can be carried out using various methods, such as presenting a photograph or video of a face, creating a

3D model of a face, or even using masks or makeup to alter the appearance of a face. These attacks pose a serious threat to the security and reliability of facial recognition systems, as they can allow unauthorized individuals to gain access to secure areas or systems or steal sensitive information or personal identity.

To address the challenges posed by face spoofing attacks, facial recognition systems typically employ facial liveness detection algorithms and other measures to ensure that the face being presented is a real, live face. Face anti-spoofing algorithms, also known as face liveness detection, refer to detecting whether a face presented for authentication is a real, live face or a fake. These algorithms have become increasingly important in recent years as facial recognition technology has grown, and with it, the potential for fraudsters to use artificial or manipulated images to bypass security systems.

The aim of this paper is to propose a modified version of majority voting that ensembles the votes of six classifiers for multiple video chunks to improve the accuracy of face anti-spoofing. The proposed method builds upon the existing face liveness detection techniques and introduces a novel approach that combines multiple classifiers and video chunks to enhance the accuracy of the system. This approach provides a robust and effective solution for detecting face spoofing attacks, which can pose a serious threat to the security of facial recognition systems.

In the following sections, the existing techniques for face anti-spoofing will be mentioned. We will then present our proposed method, including the details of the modified majority voting approach, and the experimental results obtained from the evaluation of the proposed method on the Replay Attack dataset. Finally, the paper will conclude with an analysis of the proposed method and its potential implications for the field of face anti-spoofing.

## 2. Related Work

Face presentation attack detection, also known as face liveness detection, is a highly active area of research within computer vision that has seen a significant increase in publications in recent years. Initially, many methods for presentation attack detection (PAD) were proposed that were based on traditional handcrafted features[2]−[5]. Most traditional algorithms were designed based on human liveness cues and handcrafted features. In terms of methods that rely on liveness cues, some examples include eye-blinking[2], face and head movement [6] (such as nodding and smiling), gaze tracking[7], [8], and remote physiological signals.

Several hybrid methods combining handcrafted features with deep learning techniques have been proposed for static and dynamic face PAD [9]−[12]. In addition, there has been a growing interest in developing end-to-end deep learning-based methods for this task [13]−[19]. These methods leverage the power of deep neural networks to learn discriminative features directly from the raw input data, allowing for more effective detection of presentation attacks. Both the hybrid and end-to-end deep learning-based methods have shown promising results in recent studies and are expected to continue to be essential research areas in face liveness detection.

Most works on face liveness detection[19]−[22] treat it as a binary classification problem, in which the goal is to distinguish between live and spoofing faces. In other words, the problem is typically framed as assigning a label of '0' for live faces and '1' for spoofing faces (or vice versa). This binary classification approach allows for a simple binary cross-entropy loss for supervised training of the model. Many face-liveness detection systems have been developed using this approach and have achieved impressive results in recent years. However, there is still room for improvement, and researchers are exploring more sophisticated methods for training and evaluating face-liveness detection models.

Unlike many other binary vision tasks, such as human gender classification, face liveness detection is a self-evolving problem in which attacks and defenses develop iteratively, making it much more challenging. In addition, gender classification and other similar tasks often rely on obvious appearance-based semantic clues, such as hairstyle, clothing, and facial shape. However, the intrinsic features of face liveness detection, such as material and geometry, are usually content-irrelevant, subtle, and contain fine-grained details that are difficult to distinguish, even for the human eye. Therefore, conventional Convolutional Neural Networks (CNNs) with a single binary loss may be able to effectively mine different kinds of semantic features for tasks like gender classification. Still, they may discover arbitrary and unfaithful clues, such as screen bezels, when used for spoofing detection in face liveness detection. As a result, researchers are exploring alternative approaches that can better capture the subtle and complex features required for accurate face liveness detection.

Despite the tremendous success of deep learning and CNNs in various computer vision tasks such as image classification [20], [21], semantic segmentation [22], and object detection [23], they still face significant challenges in the context of face liveness detection. One of the main challenges is the overfitting problem, which arises due to the limited amount and diversity of training data available for face liveness detection. Since the variations in presentation attacks are numerous and can be subtle, it isn't easy to collect a sufficiently large and diverse dataset to train the model effectively. This limited data availability can lead to overfitting, where the model becomes too specialized to the training data and fails to generalize to new, unseen data. As a result, researchers are exploring various strategies to address this problem, such as data augmentation, transfer learning, and domain adaptation. These techniques improve the model's generalization performance by augmenting the training data or leveraging information from related tasks or domains.

Handcrafted features, such as Local Binary Patterns (LBP) [24] , Histogram of Oriented Gradients (HOG) [25] descriptors, image quality [26] , optical flow motion [27] , and remote photoplethysmography (rPPG) clues [28], are highly discriminative in distinguishing real faces from presentation attacks. As a result, some recent works have proposed hybrid approaches that combine handcrafted features with deep features for face liveness detection. By leveraging the strengths of both techniques, these hybrid models can achieve better performance than using either method alone. For example, handcrafted features can capture subtle clues that deep features may miss. In contrast, deep features can learn more complex and abstract representations that are difficult to engineer manually. These hybrid methods have shown promising results in recent years and are a promising direction for improving the accuracy and robustness of face liveness detection.

Some face liveness detection approaches adopt a two-step process, where handcrafted features are first extracted from the face inputs. Then CNNs are employed for semantic feature representation (see Figure 1(a) for paradigm). This approach takes advantage of the strengths of handcrafted features and deep learning methods.

For example, in one approach proposed by Cai and Chen [29] , color texture-based static features are extracted from each frame using multi-scale color LBP features as local texture descriptors. These features are then fed into a random forest classifier for semantic representation. Similarly, Khammari [12] combines LBP and Weber local descriptor encoded CNN features to preserve the local intensity and edge orientation information. However, since local descriptor-based features lose pixel-level details compared to the original face input, the model's performance may be limited.

In addition to static features, dynamic features such as motion, illumination changes, and physiological signals can also be efficient inputs for CNNs. For example, Feng et al. [30] propose to train a multi-layer perceptron using dense optical flow-based motions extracted from temporal frames, which reveal anomalies in print attacks. By incorporating static and dynamic features, these hybrid approaches can improve the accuracy and robustness of face liveness detection.

Li et al. [31] proposed a method to capture abnormal reflection changes commonly found in replay attacks. The method involves using a one-dimensional convolutional neural network (1D CNN) with inputs of the intensity difference histograms obtained from reflectance images. The proposed method can effectively distinguish between live faces and replay attacks by analyzing the intensity differences between video frames. Furthermore, using 1D CNN enables the model to learn the temporal information across frames and capture the subtle changes in illumination caused by replay attacks.

Several other face liveness detection methods follow the hybrid framework shown in Figure 1(b), where handcrafted features are extracted from deep convolutional features. For example, Li et al. [35] utilized the block principal component analysis (PCA) technique to filter out deep irrelevant features to reduce the redundancy of face liveness detection-unrelated information. Then, the remaining informative features were concatenated with handcrafted features like LBP and color histogram to represent the final feature. Similarly, Huang et al. [32] proposed a hybrid face liveness detection approach that combines handcrafted features, including LBP, Gabor filter, and color histograms, with deep features extracted from a pre-trained CNN. The proposed method achieved better performance than using handcrafted or deep features alone. Additionally, Wang et al. [33] proposed a deep reinforcement learning-based face liveness detection approach, which uses handcrafted features to guide the learning of a deep neural network for face liveness detection. The proposed method achieved state-of-the-art performance on several benchmark datasets.

Researchers have recently extended their focus beyond static spoof patterns and explored handcrafted dynamic temporal clues for face anti-spoofing using well-trained deep models. Asim et al. [10] extracted deep dynamic textures using the LBP of Three Orthogonal Planes (LBP-TOP) from sequential convolutional features, while Shao et al. [32] utilized optical flow to extract motion features from the same type of features. However, one limitation of this hybrid framework is that the effectiveness of the handcrafted features largely depends on the well-trained convolutional features. It is still unclear whether shallow or deep convolutional features are more suitable for different types of handcrafted features, and further research is required to explore this aspect. [33]

Hybrid frameworks that combine handcrafted and deep convolutional features have gained popularity due to their ability to leverage the strengths of both types of features. In Figure 1(c), a popular approach is to fuse handcrafted and deep features for a more generic representation. For example, harifi [34] proposed using the predicted scores from handcrafted LBP features and deep VGG16 model for more reliable predictions. However, it is challenging to determine the optimal score weights for these two kinds of features. Another approach Rehmana et al. [10], [39] proposed using HOG and LBP maps to perturb and modulate the low-level convolutional features. While including local prior knowledge from handcrafted features enhances the discriminative capacity of the model, it may suffer from semantic representation degradation.

For temporal methods, Li et al. [35] extract intensity variation features through a 1D CNN and fuse them with motion blur features from motion-magnified face videos to detect replay attacks. This approach leverages the dynamic discrepancy between bonafide and PAs and can detect replay attacks based on the intensity variations caused by abnormal reflection changes. However, one limitation of these hybrid frameworks is that the handcrafted features depend highly on the well-trained convolutional features. Furthermore, it is still unknown which type of convolutional feature is more suitable for different kinds of handcrafted features.

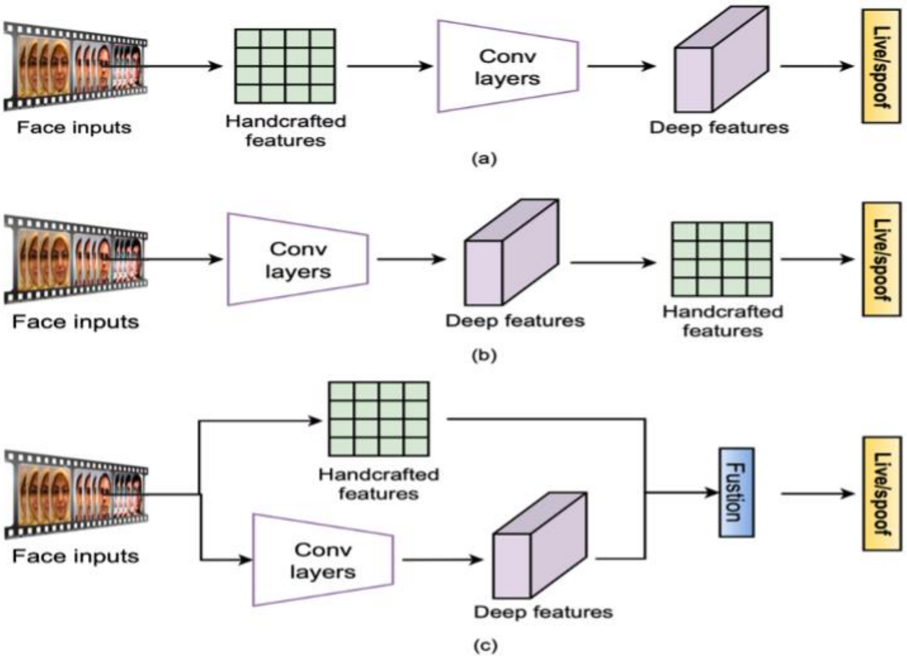

*Figure 1. Hybrid Frameworks for Face Anti-spoofing*
*(a) Deep features from handcrafted features; (b) Handcrafted features from deep features; (c) Fused handcrafted and deep features.[36]*

Deep learning methods have gained much attention recently due to their ability to automatically learn feature representations from raw data, leading to state-of-the-art results in various applications, including face anti-spoofing. However, traditional machine learning methods, such as support vector machines (SVMs) and random forests, can still achieve good results with much lower computational requirements. In our work on face anti-spoofing using frame difference analysis, we utilized a Restrictive Voting classifier for feature representation with an outstanding performance. Our approaches highlight the potential of traditional machine learning methods for face anti-spoofing, particularly in scenarios where computational resources are limited.

## 3. DATASET

Replay-Attack [1] is used in our experiments. The Replay-Attack Database for face spoofing consists of 1300 video clips of photo and video attack attempts to 50 clients under different lighting conditions. All videos are generated by having a (real) client trying to access a laptop through a built-in webcam or by displaying a photo or a video recording of the same client for at least 9 seconds.

## 4. Methodology

The proposed algorithm consists of two phases: building up the Artificial Intelligence (AI) models used in the restrictive voting algorithm and then using these AI models to construct the restrictive voting algorithm.

### *4.1 Building ML Models*

### *4.1.1 Dataset Pre-processing*

In this study, a given dataset contains multiple videos with varying lengths. To effectively analyze these videos, a series of steps must be taken, as depicted in Figure 2. One such step involves the application of a frame sampling technique, which extracts sub-clips that are two seconds in length with a one-second overlap between successive sub-clips. This approach reduces the overall amount of data that needs to be processed while retaining important information. Each of these samples is treated as a separate video for our analysis. Then each sample's color was transformed by converting them from the RGB (Red, Green, Blue) color space to the grayscale format. This transformation serves several purposes. Firstly, it reduces the amount of visual noise and distractions present in the original RGB format, allowing us to focus on the more important features of the video. Secondly, it allows us to extract more meaningful information from the videos. Finally, the grayscale format removes the chromatic variations in the original color space and reduces the data dimensionality.

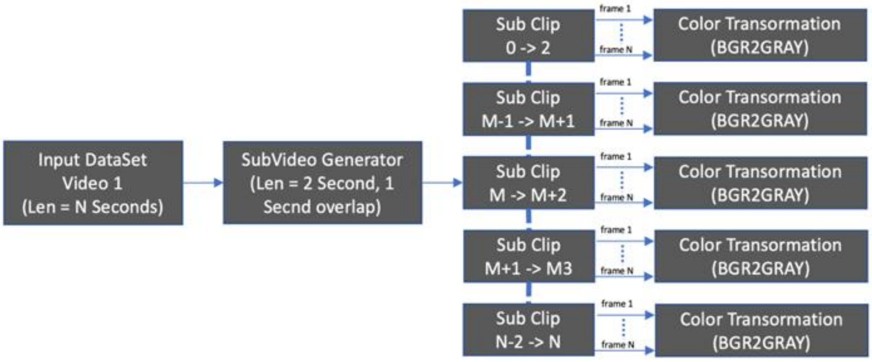
Input DataSet Video 1 (Len = N Seconds)
SubVideo Generator (Len = 2 Second, 1 Secnd overlap)
Sub Clip 0 -> 2
Sub Clip M-1 -> M+1
Sub Clip M -> M+2
Sub Clip M+1 -> M3
Sub Clip N-2 -> N
frame 1
frame N
Color Transormation (BGR2GRAY)

*Figure 2. Single Video Pre-processing*

### *4.1.2 Feature Extraction*

The resulting video consists of N frames, and the differences between each consecutive frame were calculated, resulting in N-1 differences. The frame difference measures how much the content of two consecutive frames differs, and it is often used in video analysis to identify changes or movements in the scene. By calculating the frame differences, Information about the dynamics of the scene can be extracted, enabling the identification of significant events or transitions. After calculating the frame differences, the LBP for each of these differences was computed. The LBP is a texture descriptor that captures the spatial arrangement of pixels in an image and is often used in image analysis to extract features relevant to a particular task. By calculating the LBP for each frame difference, important features could be extracted from the video that may be relevant to our analysis. Finally, all these local binary patterns will be aggregated into a single feature vector (Figure 3), which can be used for further analysis or as input to a machine-learning model. This feature vector represents a compact and comprehensive video representation and can be used to extract meaningful information about its content. In summary, this process enables us to effectively extract and analyze important features from the videos in our dataset, providing us with a better understanding of their content and potentially allowing us to classify or classify them in some way. By applying frame sampling, color transformation, frame difference calculation, and LBP computation, a rich set of features could be extracted from the videos that can be used for various purposes.

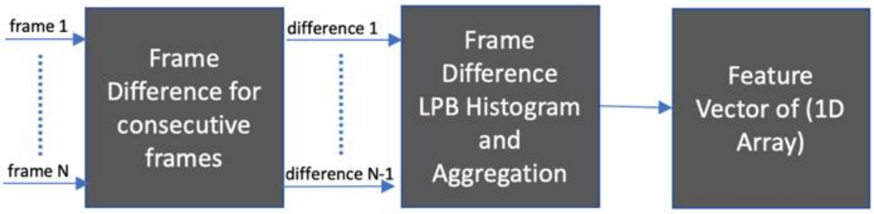
frame 1
frame N
Frame Difference for consecutive frames
difference 1
difference N-1
Frame Difference LPB Histogram and Aggregation
Feature Vector of (1D Array)

*Figure 3. Single Sub-clip Feature Extraction*

### *4.2 Building Multiple AI Models*

The training dataset consisted of 360 training samples; after applying the frame sampling resulting in 3240 sub-clip, extract feature vector for each sup-clip to be input for this phase to train multiple AI models. Multiple classifiers were used to identify the liveness of faces, focusing on minimizing the number of false positives in the confusion matrix. Multiple standalone models were built using various classifiers, including SVM with RBF kernel, Nearest Neighbors, Gaussian Process, Decision Tree, Adaboost, and Majority Voting. Each model was trained to identify real or fake faces alone (see Figure 4). SVM with RBF kernel is a popular approach in machine learning due to its ability to handle non-linearly separable data. The SVM algorithm finds the best hyperplane that separates different classes in a high-dimensional feature space by maximizing the margin, the distance between the closest data points of different classes. The RBF kernel is a popular choice for SVM because it can handle non-linear decision boundaries by projecting the data into a higher-dimensional space.

Nearest Neighbors is a simple yet effective approach in machine learning, particularly for classification tasks. The Nearest Neighbors algorithm classifies a new data point based on the class of its closest neighbors in the feature space. The method is based on a simple distance metric, such as Euclidean distance, to determine the similarity between data points. One of the main advantages of Nearest Neighbors is its ability to handle non-linearly separable data and its robustness to noise and outliers. Gaussian Process is a probabilistic machine learning approach that can handle classification and regression tasks. Gaussian Process models the underlying distribution of the data as a multivariate Gaussian distribution and can make predictions about unseen data points based on their probability distribution. In addition, Gaussian Process can model complex and non-linear relationships between input features and output labels using a kernel function, which defines the similarity between data points in the feature space. Decision Tree is a popular approach in machine learning, particularly for classification and regression tasks. The Decision Tree algorithm recursively splits the data into subsets based on the values of the input features and assigns a class label or a predicted value to each leaf node of the tree. The decision tree is constructed by selecting the feature and the threshold value that maximizes the reduction in impurity at each step. AdaBoost is an ensemble approach in machine learning that can improve the performance of a weak classifier by iteratively adjusting the weights of the training instances. The idea behind AdaBoost is to train a sequence of weak classifiers iteratively, and each classifier focuses on the misclassified instances of the previous classifier. Finally, Majority Voting is an ensemble approach in machine learning that combines the predictions of multiple classifiers to improve the overall performance. Majority Voting is a simple ensemble method that uses a majority voting rule to make the final prediction. The idea behind Majority Voting is that combining the predictions of multiple classifiers will make the final predictions more accurate and robust.

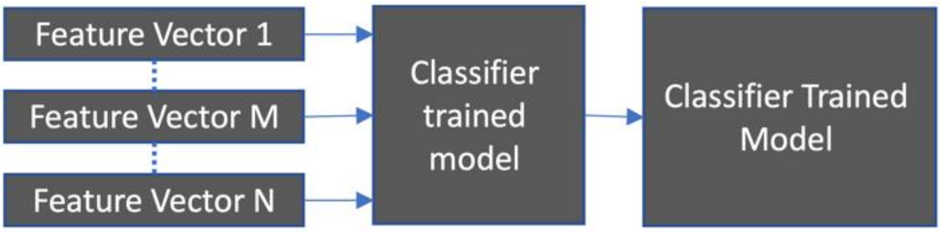

*Figure 4. AI Model Training*

### *4.3 Testing the AI Model Accuracy*

The testing dataset consisted of 480 testing samples, and after applying the frame sampling resulting in 4320 sub-clip and extract feature vector for each sup-clip to be input for this phase to test each trained AI model alone (Figure 5).

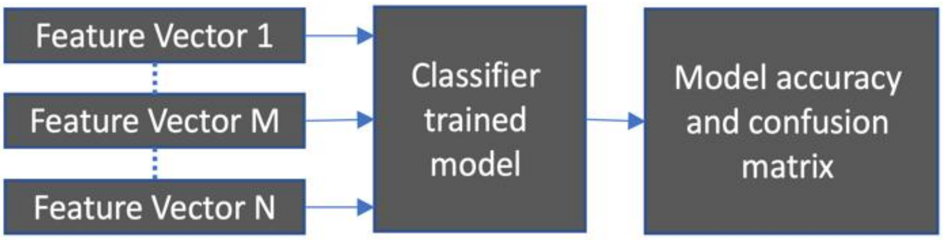

*Figure 5. AI model Testing*

### *4.4 Building Restrictive Voting*

In this study, the primary concern is to reduce the number of false positive recognized samples to zero, while keeping the false negative recognized samples minimal. To achieve this, a machine learning classifier was developed called Restrictive Voting based on ensembling multiple classifiers. There are many techniques for constructing ensembles, and in this case, various classifiers were employed to create the proposed Restrictive Voting classifier. Each classifier used to build the restrictive voting classifier helped to improve the accuracy, especially the number of false positive recognized samples. Restrictive voting builds its predictions by ensembling the weighted predictions of forming classifiers and thresholding the accumulative weighted predictions as depicted in Figure 6. Each model from the forming models worked on its own OMP thread deciding its predictions for overall sub-videos. After all, threads complete their work, and another thread combines the weighted predictions of all classifiers for each sub-videos belonging to one full video across all models and thresholds the combined predictions by 65% to decide the class of the full video as equations 1 and 2 show. The decision to use a threshold of 65% was made to further minimize the FAR. This threshold was selected after an iterative method was employed to evaluate the system's performance under different threshold values, and it was found that setting the threshold at 65% resulted in a zero FAR. This threshold enables the system to effectively prevent unauthorized access by impostors.

$$\mathrm{WAC} = \frac{\sum_{i=0}^{M}\sum_{j=0}^{N} w(i) * c(i,j)}{\sum_{i=0}^{M} w(i)} \tag{1}$$

Equation 1 calculates the weighted average of the classifications (WAC) from M classifiers for N sub-videos. Here is an explanation of the variables and parameters in the equation:

M is the number of classifiers

N is the number of sub-videos

w(i) is the weight of classifier i, where i ranges from 0 to M-1

c(i,j) is the classification of classifier i for sub-video j, where i ranges from 0 to M-1 and j ranges from 0 to N-1

WAC is the weighted average classification.

$$\mathrm{C} = \begin{cases} 1, & \text{if WAC} > 0.65 \\ 0, & \text{oth erwise} \end{cases} \tag{2}$$

In equation 2, C is the final classification computed based on the value of WAC. If WAC is greater than 0.65, then C is set to 1. Otherwise, C is set to 0.

Ensemble methods are effective in various machine learning tasks. Our Restrictive Voting classifier will likely be particularly useful when the individual classifiers are diverse and have complementary strengths. Furthermore, by combining the decisions of multiple classifiers, we can take advantage of these models' collective knowledge and expertise, which can result in improved performance compared to using a single classifier. Overall, our Restrictive Voting classifier represents a promising approach to machine learning that is likely to be useful in a wide range of applications. By ensembling the predictions of multiple classifiers, a more robust and reliable model could be achieved that can handle complex and varied data sets.

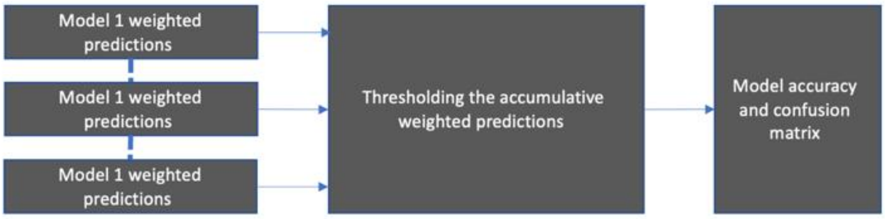
Model 1 weighted predictions
Model 1 weighted predictions
Model 1 weighted predictions
Thresholding the accumulative weighted predictions
Model accuracy and confusion matrix

*Figure 6. Restrictive Voting*

The flowchart represented in Figure 7 shows how the Restrictive Voting algorithm works.

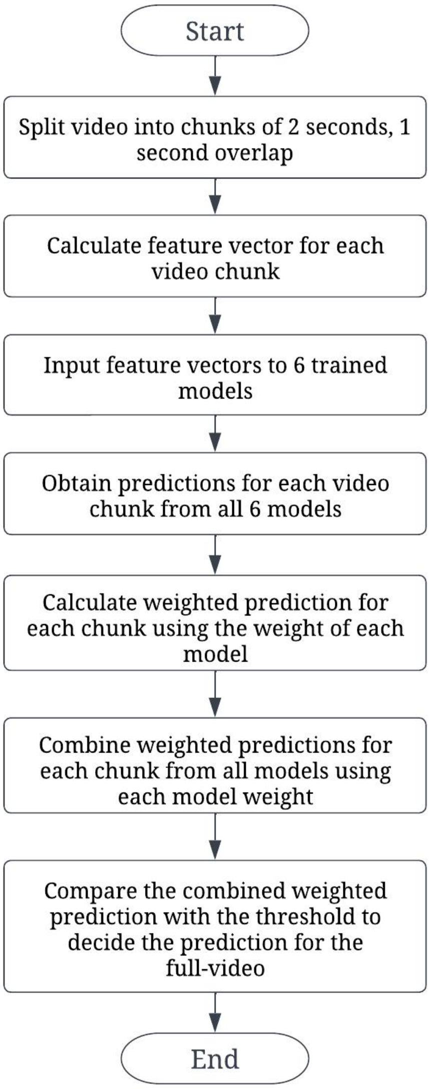
Start
Split video into chunks of 2 seconds, 1 second overlap
Calculate feature vector for each video chunk
Input feature vectors to 6 trained models
Obtain predictions for each video chunk from all 6 models
Calculate weighted prediction for each chunk using the weight of each model
Combine weighted predictions for each chunk from all models using each model weight
Compare the combined weighted prediction with the threshold to decide the prediction for the full-video
End

*Figure 7. Restrictive Voting Flowchart*

## 5. Results

In this study, the proposed Restrictive Voting technique was compared with existing state-of-the-art methods on the Replay-Attack data set. The Replay-Attack Database for face spoofing is comprised of 1300 video clips that have been generated using two methods. Firstly, by having a real client attempting to access a laptop through a built-in webcam, and secondly, by displaying a photo or video recording of the same client for a minimum of 9 seconds. The resulting videos are in color and have a resolution of 320 pixels (width) by 240 pixels (height), and were recorded on a Macbook laptop using the QuickTime framework with a Motion JPEG codec, saved in ".mov" format, and have a frame rate of approximately 25 Hz. These video clips are divided into four categories, namely, training, validation, testing, and enrollment. For this study, only the training and testing categories were utilized. The training set contains a total of 360 video clips, consisting of 60 real-access attempts and 300 spoofing attempts performed under various lighting conditions. The test set, on the other hand, contains 480 video clips, comprising 80 real-access attempts and 400 spoofing attempts performed under different lighting conditions. Our evaluation was based on two performance metrics: HTER and EER (see Table 1). HTER is a composite metric that quantifies the average of the system′s FAR and the False Rejection Rate (FRR).

*Table 1. EER (%) and HTER (%) on Replay-Attack.*

| **Method** | **EER (%)** | **HTER (%)** |
|---|---|---|
| **Fine-tuned VGG-Face**[37] | 8.40 | 4.30 |
| **DPCNN**[37] | 2.90 | 6.10 |
| **Multi-Scale**[38] | 2.14 | - |
| **YCbCr+HSV-LBP**[39] | 0.40 | 2.90 |
| **Fisher Vector**[40] | 0.10 | 2.20 |
| **Moire pattern**[41] | - | 3.30 |
| **Patch-based CNN**[17] | 4.44 | 3.78 |
| **Depth-based CNN**[17] | 3.78 | 2.52 |
| **Patch&Depth Fusion**[17] | 0.79 | 0.72 |
| **FASNet**[42] | - | 1.20 |
| **Ours (Restrictive Voting)** | 2.5 | 2.75 |

Mathematically, HTER is calculated as shown in equation 3:

$$\mathrm{HTER} = \frac{\mathrm{FAR} + \mathrm{FRR}}{2} \tag{3}$$

Our analysis suggests that HTER and EER may not be the most appropriate metrics for evaluating the effectiveness of a face spoofing attack detection system. As these metrics provide a balanced assessment of the system's accuracy for legitimate users and impostors. Given the critical importance of preventing unauthorized access by impostors, we believe it is essential to focus primarily on reducing the FAR while putting our eyes on securing a small HTER in the second place. In line with this, the threshold was raised to 65%, resulting in zero FAR, and compared the performance of our proposed Restrictive Voting method, which utilizes multiple forming classifiers with individual classifiers. The Restrictive Voting method combines an SVM with an RBF kernel, a Nearest Neighbors classifier, a Gaussian Process, a Decision Tree, AdaBoost, and Majority Voting classifier. The FAR and HTER were evaluated for each classifier. Then, compared them to the results obtained with the Restrictive Voting method, which allowed us to assess the effectiveness of ensembling multiple classifiers in the Restrictive Voting approach. Our evaluation results in Table 2 demonstrate that our proposed Restrictive Voting method outperforms each standalone classifier regarding both FAR and HTER.

*Table 2. FAR (%) and HTER (%) on Replay-Attack with 0.65 Thresholds.*

| Classifier | FAR (%) | HTER (%) |
|---|---|---|
| **SVM-RBF** | 1.25 | 5.625 |
| **NearestNeighbors** | 1.5 | 11.375 |
| **Gaussian Process** | 0 | 50 |
| **Decision Tree** | 0 | 10.625 |
| **AdaBoost** | 1.25 | 22.5 |
| **Majority Voting** | 0.25 | 7.625 |
| **Ours (Restrictive Voting)** | 0 | 6.25 |

Furthermore, though the proposed method has a similar FAR with several classifiers, the HTER is smaller than each standalone classifier, so the proposed method has lower false acceptance and rejection rates.

Figure 8 shows that using the restrictive voting technique affected the overall accuracy positively. The number of false positive recognized samples was zero while maintaining a small number of false negative recognized samples. The Restrictive Voting classifier has a testing accuracy of approximately 97.9% with 0 false positive samples, 10 false negative samples, 70 true positive samples 400 true negative samples.

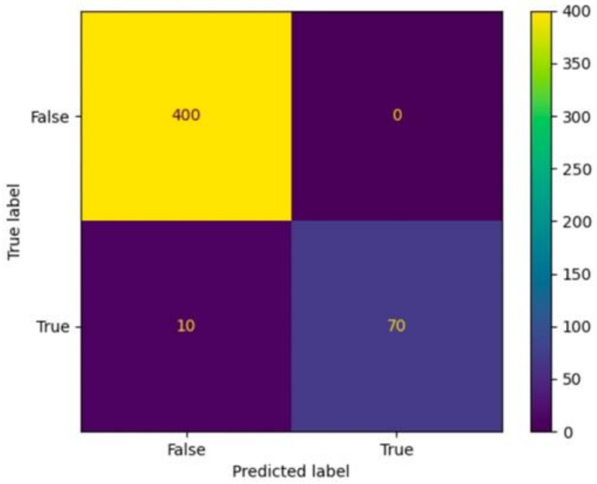

*Figure 8. Restrictive Voting*

## 6. Conclusions

This paper presented a video processing pipeline to identify face spoofing attacks using six traditional machine learning algorithms and a modified version of the majority voting algorithm called Restrictive Voting. The result of this algorithm shows zero FAR, which is most suitable for sensitive and mission-critical applications. Also, the comparison of other complex deep learning algorithms against the proposed algorithm shows a comparable result despite the simplicity of the proposed algorithm. This work can be a cornerstone for further investigation with other machine learning techniques that can act as a feature selection mechanism for complex scenarios.

## References

[1] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany, *IEEE Xplore*, 2012, Available: https://ieeexplore.ieee.org/document/6313548

[2] J. W. Li, “Eye blink detection based on multiple Gabor response waves,” *Semantic Scholar*, 2008, Available: https://www.semanticscholar.org/paper/Eye-blink-detection-based-on-multiple-Gabor-waves-Li/d8f6be181f6d8f508e77d73b4cb1284c27f5e178

[3] X. Li, J. Komulainen, G. Zhao, P. C. Yuen et al., “Generalized face anti-spoofing by detecting pulse from face videos — Hong Kong Baptist University,” 2016, Available: https://scholars.hkbu.edu.hk/en/publications/generalized-face-anti-spoofing-by-detecting-pulse-from-face-video-2

[4] T. de Freitas Pereira, A. Anjos, J. M. de Martino, and S. Marcel, “LBP-TOP based countermeasure against face spoofing attacks,” *Computer Vision - ACCV 2012 Workshops*, vol. 7728 LNCS, no. PART 1, pp. 121–132, 2013, doi: 10.1007/978-3-642-37410-4_11.

[5] K. Patel, H. Han, and A. K. Jain, “Secure Face Unlock: Spoof Detection on Smartphones,” *IEEE Journals & Magazine*, vol. 11, no. 10, pp. 2268–2283, 2016, Available: https://ieeexplore.ieee.org/document/7487030

[6] J. Bigun, H. Fronthaler et al., “A liveness detection method for face recognition based on optical flow field,” *IEEE Xplore*, 2004, Available: https://ieeexplore.ieee.org/document/5054589

[7] J. Bigun, H. Fronthaler et al., “Assuring liveness in biometric identity authentication by real-time face tracking,” Proceedings of the 2004 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety, Venice, Italy, *IEEE Xplore*, pp. 104-111, 2004, Available: https://ieeexplore.ieee.org/document/1360218

[8] A. Ali, F. Deravi, and S. Hoque, “Liveness Detection Using Gaze Collinearity,” In 2012 Third International Conference on Emerging Security Technologies, Lisbon, Portugal, *IEEE Xplore*, pp. 62–65, 2012, Available: https://ieeexplore.ieee.org/abstract/document/6328083/authors#authors

[9] X. Song, X. Zhao, L. Fang, and T. Lin, “Discriminative representation combinations for accurate face spoofing detection,” *Pattern Recognit*, vol. 85, pp. 220–231, Jan. 2019, doi: 10.1016/J.PATCOG.2018.08.019.

[10]M. Asim, Z. Ming, and M. Y. Javed, “CNN based spatio-temporal feature extraction for face anti-spoofing,” *IEEE Xplore*, 2017, Available: https://ieeexplore.ieee.org/abstract/document/7984552

[11]Y. A. U. Rehman, L. M. Po, and J. Komulainen, “Enhancing deep discriminative feature maps via perturbation for face presentation attack detection,” *Image Vis Comput*, vol. 94, no. 103858, Feb. 2020, doi: 10.1016/J.IMAVIS.2019.103858.

[12]M. Khammari, “Robust face anti-spoofing using CNN with LBP and WLD,” *IET Image Process*, vol. 13, no. 11, pp. 1880–1884, Sep. 2019, doi: 10.1049/IET-IPR.2018.5560.

[13]Y. Liu, A. Jourabloo, and X. Liu, “Learning Deep Models for Face Anti-Spoofing: Binary or Auxiliary Supervision,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 389–398, 2018.

[14]Z. Yu et al., “Searching Central Difference Convolutional Networks for Face Anti-Spoofing,” pp. 5295–5305, 2020. Accessed: Feb. 28, 2023. [Online]. Available: https://github.com/ZitongYu/CDCN.

[15]Z. Yu, X. Li, X. Niu, J. Shi, and G. Zhao, “Face Anti-Spoofing with Human Material Perception,” *Lecture Notes in Computer Science*, vol. 12352 LNCS, pp. 557–575, 2020, doi: 10.1007/978-3-030-58571-6_33/COVER.

[16]X. Yang et al., “Face Anti-Spoofing: Model Matters, so Does Data.” pp. 3507–3516, 2019.

[17]Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, “Face anti-spoofing using patch and depth-based CNNs,” IEEE International Joint Conference on Biometrics, IJCB 2017, vol. 2018-January, pp. 319–328, Jan. 2018, doi: 10.1109/BTAS.2017.8272713.

[18]Z. Yu et al., “Multi-Modal Face Anti-Spoofing Based on Central Difference Networks.” pp. 650–651, 2020. Accessed: Feb. 28, 2023. [Online]. Available: https://github.com/ZitongYu/CDCN.

[19]S. Zhang et al., “CASIA-SURF: A Large-Scale Multi-Modal Benchmark for Face Anti-Spoofing,” *IEEE Trans Biom Behav Identity Sci*, vol. 2, no. 2, pp. 182–193, Apr. 2020, doi: 10.1109/TBIOM.2020.2973001.

[20]K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016. Accessed: Feb. 28, 2023. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[21]G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, 2017. Accessed: Feb. 28, 2023. [Online]. Available: https://github.com/liuzhuang13/DenseNet.

[22]J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2015.

[23]S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Adv Neural Inf Process Syst*, vol. 28, 2015, Accessed: Feb. 28, 2023. [Online]. Available: https://github.com/

[24]T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans Pattern Anal Mach Intell*, vol. 28, no. 12, pp. 2037–2041, 2006, doi: 10.1109/TPAMI.2006.244.

[25]N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", Accessed: Feb. 28, 2023. [Online]. Available: http://lear.inrialpes.fr

[26]J. Galbally and S. Marcel, "Face anti-spoofing based on general image quality assessment," *Proceedings of International Conference on Pattern Recognition*, pp. 1173–1178, Dec. 2014, doi: 10.1109/ICPR.2014.211.

[27]T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans Pattern Anal Mach Intell*, vol. 33, no. 3, pp. 500–513, 2011, doi: 10.1109/TPAMI.2010.143.

[28]X. Niu, Z. Yu, H. Han, X. Li, S. Shan, and G. Zhao, "Video-Based Remote Physiological Measurement via Cross-Verified Feature Disentangling," *Lecture Notes in Computer Science* (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12347 LNCS, pp. 295–310, 2020, doi: 10.1007/978-3-030-58536-5_18/COVER.

[29]R. Cai and C. Chen, "Learning deep forest with multi-scale Local Binary Pattern features for face anti-spoofing," Oct. 2019, doi: 10.48550/arxiv.1910.03850.

[30]L. Feng et al., "Integration of image quality and motion cues for face anti-spoofing: A neural network approach," *J Vis Commun Image Represent*, vol. 38, pp. 451–460, Jul. 2016, doi: 10.1016/J.JVCIR.2016.03.019.

[31]L. Li, Z. Xia, X. Jiang, Y. Ma, F. Roli, and X. Feng, "3D face mask presentation attack detection based on intrinsic image analysis," *IET Biom*, vol. 9, no. 3, pp. 100–108, May. 2020, doi: 10.1049/IET-BMT.2019.0155.

[32]R. Shao, X. Lan, and P. C. Yuen, "Joint discriminative learning of deep dynamic textures for 3D mask face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 923–938, Apr. 2019, doi: 10.1109/TIFS.2018.2868230.

[33]G. Zhao and M. Pietikäinen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," IEEE Trans Pattern Anal Mach Intell, vol. 29, no. 6, pp. 915–928, Jun. 2007, doi: 10.1109/TPAMI.2007.1110.

[34]O. Sharifi, "Score-Level-based Face Anti-Spoofing System Using Handcrafted and Deep Learned Characteristics," *Image, Graphics and Signal Processing*, vol. 2, pp. 15–20, 2019, doi: 10.5815/ijigsp.2019.02.02.

[35]L. Li, Z. Xia, A. Hadid, X. Jiang, H. Zhang, and X. Feng, "Replayed video attack detection based on motion blur analysis," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 9, pp. 2246–2261, Sep. 2019, doi: 10.1109/TIFS.2019.2895212.

[36]Z. Yu, Y. Qin, X. Li, C. Zhao, Z. Lei, and G. Zhao, "Deep Learning for Face Anti-Spoofing: A Survey," *IEEE Trans Pattern Anal Mach Intell*, Jun. 2021, doi: 10.48550/arxiv.2106.14948.

[37]L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li, and A. Hadid, "An original face anti-spoofing approach using partial convolutional neural network," In *2016 6th International Conference*

*on Image Processing Theory, Tools and Applications*, IPTA 2016, Jan. 2017, doi: 10.1109/IPTA.2016.7821013.

[38]J. Yang, Z. Lei, and S. Z. Li, "Learn Convolutional Neural Network for Face Anti-Spoofing," Aug. 2014, doi: 10.48550/arxiv.1408.5601.

[39]Z. Boulkenafet, J. Komulainen, and A. Hadid, "face anti-spoofing based on color texture analysis," *Proceedings of International Conference on Image Processing*, ICIP, vol. 2015-December, pp. 2636–2640, Nov. 2015, doi: 10.48550/arxiv.1511.06316.

[40]Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face antispoofing using speeded-up robust features and fisher vector encoding," *IEEE Signal Process Lett*, vol. 24, no. 2, pp. 141–145, Feb. 2017, doi: 10.1109/LSP.2016.2630740.

[41]K. Patel, H. Han, A. K. Jain, and G. Ott, "Live face video vs. spoof face video: Use of moiré patterns to detect replay video attacks," *2015 International Conference on Biometrics (ICB)*, pp. 98–105, Jun. 2015, doi: 10.1109/ICB.2015.7139082.

[42]O. Lucena, A. Junior, V. Moia, R. Souza, E. Valle, and R. Lotufo, "Transfer Learning Using Convolutional Neural Networks for Face Anti-spoofing," *Lecture Notes in Computer Science*, vol. 10317, pp. 27–34, 2017, doi: 10.1007/978-3-319-59876-5_4.