Date of publication MMM-dd, yyyy, date of current version MMM-dd, yyy
www.computingonline.net/ computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.19.3.XX

Performance Comparison of Classification Algorithms for Face Anti-Spoofing using Codebook Features

SWAPNIL R. SHINDE[1,4], SUDEEP D. THEPADE[2],ANUPKUMAR M. BONGALE[3]

[1]Department of Computer Science and Information Technology, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Lavale,Pune,Maharashtra, India (e-mail: swapnil.shinde.phd2019@sitpune.edu.in)
[2]Department of Computer Engineering,Pimpri Chinchwad College of Engineering,Pune,Maharashtra,India (email: sudeepthepade@gmail.com
[3]Department of Artificial Intelligence and Machine Learning,Symbiosis Institute of Technology, Symbiosis International (Deemed University), Lavale,Pune,Maharashtra, India (e-mail: anupkumar.bongale@sitpune.edu.in)
[4]Convergytics Solutions Pvt Ltd. Bangalore,India

Corresponding author: Swapnil R. Shinde (e-mail: swapnil.shinde.phd2019@sitpune.edu.in).

This paragraph of the first footnote will contain support information, including sponsor and financial support acknowledgment. For example, “This work was supported in part by the U.S. Department of Commerce under Grant BS123456.”

ABSTRACT Face recognition systems are prone to break by using face images and video or mask methods, termed face spoofing attacks. The 2D attacks include fake photo attacks, warped photos, video display attacks, and 3D attacks performed using 3D masks. Detection of attacks with higher efficiency remains a problem due to factors such as illumination and dataset variations. The paper focuses on designing a system to detect 3D mask attacks with higher efficiency and lower error rate. The proposed system consists of use of codebook features obtained using Linde-Buzo-Gray (LBG) Algorithm and Kekre’s Error Vector Rotation (KEVR) algorithms for different sizes from 8 to 256. The results are obtained for various Machine Learning(ML) classifiers and evaluated using Attack Presentation Classification Error Rate (APCER), Half Total Error Rate (HTER) and Bonafide Presentation Classification Error Rate (BPCER) for both Algorithms on 3D MAD Dataset. The KNN variants perform well for KEVR features, and SVM with Logistic Regression has higher results for LBG features. The analysis indicates the proposed method’s improved performance over the existing methods of face anti-spoofing.

KEYWORDS Biometrics Authentication; Face Anti-Spoofing; LBG algorithm; KEVR algorithm; Machine Learning(ML) Algorithms; Presentation Attack Detection

I. INTRODUCTION

Security is a crucial factor in today’s world all over the globe; security can be implemented for the systems to achieve different goals and secure them from the outside world. Authentication is the primary goal achieved using passwords, tokens, and biometrics methods. Biometrics [1] is considered the most secure and robust method for authenticating an individual as it’s simple to implement and difficult to break; it includes mainly two mechanisms, viz. Physiological and Behavioral. Physiological traits include the face, fingerprint, retina, iris, palm geometry, etc. Face Recognition systems use Authentication systems use Face Images [2] for authentication; these face identification systems are susceptible to presentation attacks. Different attack methods for face spoofing attacks exist, primarily categorized as 2D & 3D attacks. The 2D & 3D attack methods include Printed photos, Image Display, 3D Mask attack [3] and Video replay attack. The 2D & 3D attacks lead to intruding on an individual’s or organization’s data, which can cause monetary losses. The attacks aim to fool the system into tampering and stealing the data, which is a serious concern to address. Attack detection mechanisms are proposed in the literature for 2D & 3D attacks, and standard performance metrics have been used to evaluate the same. The 3D Attack detection uses different techniques to extract relevant and useful features from face images with other State Of The Art (SOTA) mechanisms. The most comprehensive approach is based on texture [4] [5] and

shape features of 3D face images. The broad categorization of the systems for 3D face presentation attack detection [6] is as hardware-based or software-based, or hybrid (software and hardware). The Local Phase Quantization (LPQ) [7] and Gaussian Mixture Model (GMM) [4] based methods have been tested on publicly available 2D and 3D attack datasets such as Oulu-Npu [8], CASAI-FASD [9], 3D MAD [10] , etc.

The above detection methods are significant and have good detection performance, but these methods are invariant to illumination changes. They do not consider the minutiae-level features of the images to generate the feature vectors. The LBG [11], [12], and KEVR [13] algorithms have been used in the proposed system for codebook generation, which represents the feature vector for the classification stage. The codebook algorithms have not been applied prior for face spoofing detection; this is the first attempt. They extract minutiae-level features from the input images result in better detection and improved performance over relevant existing methods.

Major contributions of the paper include:

1) Application of Vector Quantization algorithms for face spoofing detection and its evaluation on the 3D MAD dataset.
2) Flattening method for conversion of codebook features to 1D array for classification.
3) Comparative analysis of various ML Algorithms based on standard parameters.

The following sections of the paper are the literature survey, proposed system, implementation, and analysis.

II. LITERATURE SURVEY

The author proposed a novel system that concatenates global & local features by extracting Binarized Statistical Image Features (BSIF) [14]. The periocular and nose region of the face is the ROI for local features extraction; this region shows variations in terms of genuine and mask face images. Global features are obtained by application of BSIF; the local and global features are obtained for color RGB and depth images. The proposed system was applied on a 3D MAD database using four different SVM classifiers. It trained them separately with a different set of features. The proposed scheme HTER is 0.03%.

The authors [15] have explored artifacts induced in the synthetic mask material during manufacturing. The proposed face spoofing system consists of 3 stages: Low-Level feature Formation, Mid-Level Extractor, and spoofing detection. Fourier spectrum is calculated from residual noise and used to generate the time spectral descriptor; this is passed for generating a visual word descriptor from the visual codebook. The authors [16] have proposed a face recognition and authentication system based on global and local features. The local features are generated with 2D DCT for the eye region, and the global features LBP and BSIF are extracted. The formula for 2D-DCT is in equation 1.

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) cos \frac{(2x+1)\pi u}{2M} cos \frac{(2y+1)\pi v}{2N} \quad (1)$$

The classification results of depth and color images are fused with the weighted sum score mechanism; the fusion score is further used for recognition.

Flash against spoofing attack [17] is an attempt that uses motion cues with hardware-based flash generation from the camera to detect real and fake attempts. In motion-based spoofing detection, [4] have considered the video and photo attack datasets to evaluate non-rigid face & rigid movements. Data-driven & cue-based technique combinations have been introduced as a novel mechanism for face liveliness detection. The steps of the system include Face key point detection, shape parameter extraction, motion extraction, and classification. The system is tested on the three publicly available 2D datasets and one 3D attack dataset. Evaluation is performed based on EER and HTER.

Quantization techniques have a focus on extracting texture features. The authors [7] have extracted seven texture-based features using different LBP, RI-LBP, BSIF, CoALBPRIC methods, LBP, LPQ, and SURF. The color space conversion is the initial stage; the images are converted to HSV and YCbCr and then used for feature extraction. Results for the seven features were analyzed for all three databases in terms of HTER. The authors have proposed a system known as Multi-Regional Convolutional Neural Networks(MRCNN) [18]; the method performs visits to local patches in an image that contains visual information and extracts information from them. The authors have explored fully convolutional network for image semantic segmentation. Three different networks are trained: global classification CNN, CNN with occlusion, and MRCNN with 3x3 output. The face images in the dataset have light radiation from right to left, so the right half is used for spoofing identification; CNN with occlusion reduces this effect. The evaluation uses HTER, FRR, FAR for three different datasets. The authors [19] have performed local and global feature extraction on the different color spaces with the application of Deep CNN [20] for feature extraction. The authors have used double-stream CNN to model relevant features from global face images and local patches obtained from HSV and YCbCr color space and integrated. The proposed system is evaluated for three cases on the CASIA [21] Replay attack dataset. The EER of fusion is very low compared to the two local and global deep feature extraction cases above. The summary of all the papers indicates that the different encoding techniques, like GMM and Fisher vector encoding, have provided key insights for useful feature extraction and improved the detection performance on different datasets. The summarization of survey papers is shown in Table 1.

Table 1. Literature Survey Table

Method Proposed	**Database Used**	**Features Extracted**	**Classification Algorithm**	**Performance evaluation parameters**	**Key findings**
Local and Global Feature Fusion Method [14]	3DMAD	Local & Global Features	SVM	HTER	Local & global features fusion improved results
Spectral and Temporal Features method [15]	Replay Attack, CASIA, 3DMAD, UVAD	Time spectral and temporal features	Partial Least square(PLS), SVM	FAR,FRR, HTER, AUC	Manufacture of mask introduces noise and artifacts that prove to be discriminating factor.
Motion Codebook with Fisher Encoding [4]	CASIA-FASD, Replay attack,3DMAD and MSU-MFSD	Shape and Texture features with their Fusion	Linear SVM(c=1)	HTER, EER,APCER	Novel motion based method with fisher encoding to build features from non-rigid and rigid movements of face.
Mutiple texture Feature extraction and fusion for HSV and YCbCr color space [7]	CASIA-FASD, Replay attack, MSU-MFSD	Texture features	Softmax Classifier	HTER,EER	Color and texture based methods perform well for display attacks but have lower performance for print attacks.
Color and Local Patch-based system with Double stream CNN [19]	CASIA-FASD, Replay Attack	Color and Deep features fusion	Softmax classifier	EER	Chrominance information are crucial in face spoofing detection along with use of CNN of different levels.
Multi-Regional CNN for Local Patches [18]	Replay Attack, OULU-NPU and SiW	Deep Features	MRCNN	Accuracy, HTER	Compared to CNN methods the MRCNN was robust to adversial attacks and traditional attacks

III. METHODOLOGY

The proposed methodology is depicted in Fig. 1. The proposed approach is implemented for two vector quantization algorithms. LBG and KEVR. The steps of the presented method are listed as follows: label=.

1) Input Color RGB image.
2) Crop face region
3) Resize the image to 64x64.
4) Apply LBG and KEVR algorithms for Codebook Generation.
5) Generate Codebooks of varying size viz. Codebook 8,16,32,64, 128,256.
6) Image Codebook of M x N is converted to one dimension of size 1xN, where N is the total no of feature values arranged sequentially for all image features.
7) Feature vector of above step is passed as input to ML classifiers SVM, KNN, and Logistic regression.
8) The KNN and SVM variants undergo feature reduction using Principal Component Analysis before classification.

The RGB image of the individual from the 3D MAD dataset is cropped to extract the face region. This image is then resized to the dimension of 64x64; resizing add's top efficiency and speeds up the feature extraction process by extracting only relevant features of the face. The resized image is passed to LBG and KEVR Algorithms for Codebook generation to generate the feature codebooks. The steps for LBG and KEVR algorithm are stated below.

Vector Quantization Algorithms: Vector Quantization(VQ) techniques are mainly used for performing data compression, it has applied for dealing with different types of data and extracting meaningful information that forms the codebook. Different research areas have applied the VQ algorithms for Content Retrieval, Image and Video detection, Biometric Recognition etc. It is a form of clustering algorithm but with some advantage over traditional methods. VQ creates codebook(CB) of different sizes by mapping the training vectors into different clusters of finite value. CB consists of N code vectors where the dimension is k. Codebooks obtained after VQ represent the feature set for the input image. Here we calculate results on different size of Code book ranging from 8 to 256.

A. LBG ALGORITHM

The LBG VQ is an recursive algorithm requiring one starting point, the first codebook C is obtained by splitting. The mean of the full training vector is used to generate the clusters to represent the first codebook which is further split into multiple halves as iteration increases. The algorithm runs with these two vectors as the initial codebook. This splitting continues further to obtain codebooks in the power of 2.

Where S - training vector obtained by sliding a window of 2x2 over the image.
K- no of clusters that represent the codebook size, such as 8,16,64, and so on.

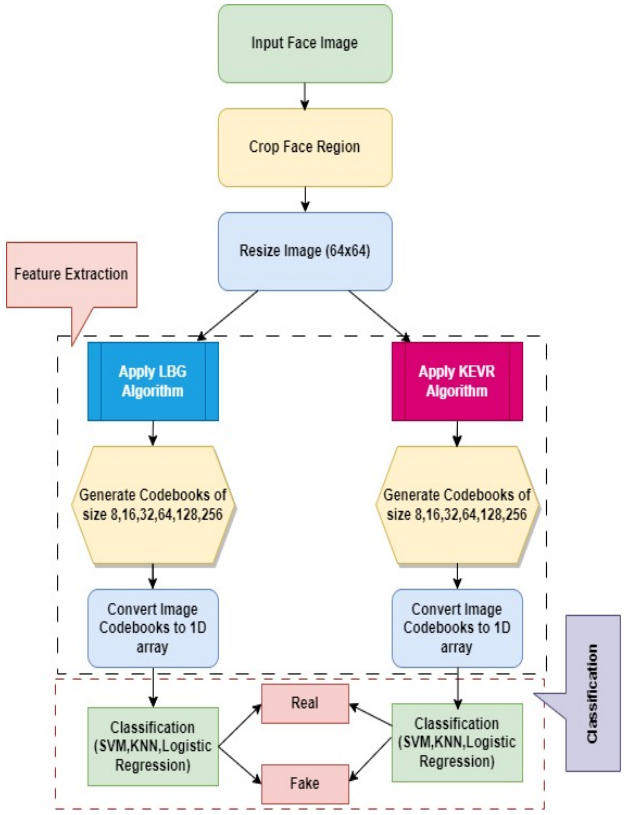

Figure 1. Proposed System Design

Algorithm 1 Algorithm for LBG [11]

0: Input training vectors S = $\{x_i \in R^d \ | i = 1, 2, , n \ \}$
0: Initiate a codebook $C = \{c_j \in R^d | j = 1, 2, , K\}$
0: Set D_0 = 0 and let k = 0
0: Classify the n training vectors into K clusters according to - $x_i \in S_q \ if ||x_i - c_q||_p \leq ||x_i - c_j||_p \ for j \neq q$
0: Update cluster centers c_j , j = 1, 2, $\cdots$, K by $c_j = \frac{1}{|s_j|} \sum x_i \in S_j x_i$
0: Set k $\leftarrow$ $k + 1$ and compute the distortion $Dk = \sum_{j=1}^{k} \sum_{x_i \in Sj} ||x_i - c_j||_p \ (D_{k-1} - D_k)/D_k > q$ (a small number) repeat steps 4 to 6
0: Output the code book $C = \{c_j \in R^d | j = 1, 2, , K\}$ =0

Rd - number of overlapping blocks obtained after using a window size of 2x2.
D - Distortion or Euclidean distance between two training vectors for cluster assignment.

Initial Codebook and Distortion are the two important aspects of this method and the training vector. In the beginning, two clusters are formed by adding some constant error to the initial code vector. Based on the Euclidean distance of the training vectors concerning the clusters, the final codebook of size 2 is obtained. This process continued til the required size codebook is not generated. The cluster elongation takes place in 135^0, which results in improper clustering; this is the drawback of LBG.

B. KEVR ALGORITHM.

Codebook generation's first step involves dividing the image into fixed-size sets, known as training vectors. The Collection of all such vectors forms the train set. The centroid of the train set code vector obtained/calculated. This centroid is then used to generate the initial two clusters by adding and subtracting the Error Vector Rotation matrix. The train vectors are then put into respective clusters based on the Euclidean distance with respect to the two code vectors c1 and c2 to form the final codebook. The vector sequence is generated by the representation of the numbers in binary form, 0 to n-1, in the n-dimensional space. The vector sequence consists of binary values obtained by replacing 0 by 1 and 1 by -1. The error vector rotation matrix used is given in Fig. 2. The Codebooks generated from the LBG and KEVR algorithm are two-dimensional matrices. The Dimension of the Codebooks is MxN for a single image, where M is the codebook size in the power of 2 and N is the intensity values of R, G, and B when a 2x2 window is applied over the image pixels. A codebook ranging from size 8 to 256 is formed for all the images in the dataset, and a combined feature vector representing the dataset is obtained.

Algorithm 2 Algorithm for KEVR Codebook generation [13]

0: Apply a 2x2 window to obtain the training vector through the mechanism of non-overlapping block division. Each obtained block represents the final train vector
0: Start with k=1
0: Calculate the centroid of the training vector set.
0: Form two vectors c1 and c2 by adding and subtracting error vector Ck from the centroid obtained above.
0: Assign train vector blocks to cluster c1 and c2 based on Euclidean distance with respect to the centroid value.
0: Compute the centroid (code vector) for clusters obtained in the above step 5 for the next cluster formation.
0: Increment k by one & iterate step 4 to step 6 for each code vector
0: Reiterate Step 3 to Step 7 to obtain the codebook of different sizes =0

The error vector rotation matrix used is given in 2
This increases the size of the feature vector; the new size is MxNxP, where M and N are the same as stated above, and P represents the no of images for which codebooks are generated. The next stage is reducing the size by the flattening process; in the proposed system, the image dimension of M x N is reduced to 1 x K i.e 1D array, where K represents the concatenation of row values of the codebook into a single row in a sequential manner. For example, if the codebook dimension is 8x12, then after flattening, the dimension will

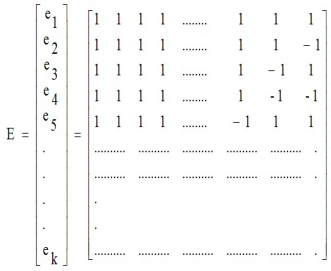

Figure 2. KEVR Error Rotation Matrix

be 1x96. The same process is applied to the MxNxP matrix to convert it to P x K. This P x K matrix is the flattened feature vector passed to the next step of classification and feature reduction. The PCA algorithm performs feature reduction by selecting the important feature values and removing the other values based on its working procedure to boost the classification performance of the model. The classification is performed using the base Supervised ML Classifiers viz. SVM, Logistic Regression, and KNN.

C. CLASSIFICATION USING MACHINE LEARNING CLASSIFIERS

The classification is performed using the MATLAB Classification Learner App using the built-in functions of SVM [22] and KNN. The KNN algorithm is applied by setting the nearest neighbor value to 5 and standardization set to 1. Similarly, the SVM function is set with a standardized linear kernel for training the model. The Classification Learner App provides the flexibility of training the models with PCA [23] enabled or disabled; the default variance for PCA in-app is set to 95 and the value set to 10. The Weighted KNN algorithm is chosen as a variant of KNN for classification and tested with enabled and disabled PCA. The Number of neighbors considered by Weighted KNN is 10 for training the feature vector. The Linear SVM has kernel scale set to automatic, Box constraint to 1, and Multiclass method as One vs. One. The classification learner app applies cross-validation with the value of k set to 5. Linear SVM has shown promising and improving results in many research work done in the past; we have used this as a base key point for applying Linear SVM on features extracted by our proposed system. The results obtained using the classifiers discussed above are satisfactory for both the LBG and KEVR features. The classification uses seven machine-learning classifiers with KNN and SVM variants and Logistic Regression.

Principal Component Analysis (PCA): -PCA [23] is a dimensionality reduction technique that identifies the important features/information from the feature set and preserves them while removing the other unwanted feature values. It performs linear dimension reduction and maps data in higher space to lower space with maximum spread. Eigenvalue Decomposition and Singular Value Decomposition(SVD) are the two main procedures used in PCA to reduce dimensionality. It selects the important eigenvectors obtained to form the principal axes, these values form the new feature set that can be passed to the classifiers.

Support Vector Machine (SVM): SVM [22] is a supervised ML technique that creates a hyperplane between two classes that best separates the binary classes. The kernels are at the heart of SVM, enabling the best hyperplane processing. SVM is applied for both regression and classification problems. There are mainly two types of SVM: 1. Linear SVM 2. Non-Linear SVM

The best hyperplane is the one with the longest range from both classes, and that's a primary aim of SVM. It follows that it will identify a variety of hyperplanes that classify the labels in such a way as to distinguish them from each other or, if at all possible, between two data points and choose one with maximum margins.

K-Nearest Neighbour(KNN): KNN [24] classifier works on the distance calculation for assignment of classes or categories for the input data. The K term defines the number of neighbours to consider for defining the exact category. The distance is calculated using the euclidean distance and input training data is assigned to a class.

IV. IMPLEMENTATION AND RESULTS

The proposed approach is implemented and tested on 3DMAD Erdogmus2013 dataset which is publicly available for testing the face presentation attack detection systems. The Entire work is implemented in the MATLAB R2016A Environment on an Intel Core i3 processor with 8GB RAM and 3.2GHZ processor. 3D Mask Attack Database is recorded in three sessions where the first two sessions are of real access and the third session is mask access. The dataset consists of images of 17 subjects and 5 videos of person per session, each video consists of 300 frames at 30 fps. The Kinect 3D sensor is used to record the samples. The database has two categories of images,real and mask images.Images from 3D MAD are shown in figure 3

Every 10th frame is analyzed and training set features are obtained for the same. This training set is passed to ML algorithms to build the models and then tested using the test set. The results are reported in terms of Accuracy , APCER, BPCER and HTER [25]. The classification is done using 16 subjects for training and 1 subject for testing which is referred to as Leave one Out manner. SVM classifier with standardized linear kernel, K-Nearest Neighbour and Logistic Regression is applied for classification and results are obtained for different variations of SVM and KNN along with Logistic Regression.

The Results of classifiers are obtained for four parameters, Accuracy , Attack Presentation Classification Error Rate(APCER) [8] and Bona-fide Presentation Classification Error Rate(BPCER) [18]. The APCER and BPCER are synonom to the FPR and FNR which can be obtained from the confusion matrix.

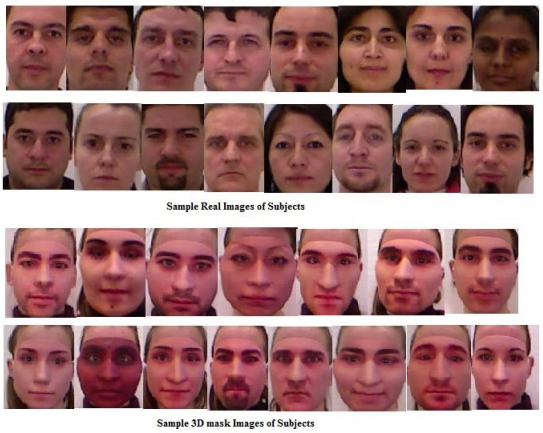

Figure 3. Sample 3D MAD Database Images [10]

The formula for all standard parameters is given in the below equations.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100 \quad (2)$$

$$BPCER = \frac{FN}{FN + TP} * 100 \quad (3)$$

$$APCER = \frac{FP}{FP + TN} * 100 \quad (4)$$

Half Total Error Rate (HTER) [26] is an important metric widely used in the biometrics domain to measure the error aspect to classify as fake or real. The Half Total Error Rate formula is given in equation 5

$$HTER = \frac{FPR + FNR}{2} * 100 \quad (5)$$

The result for classification accuracy for different ML algorithms in terms of different codebook size using KEVR algorithm is shown in Fig. 4 and for LBG algorithm its shown in Fig. 5.

APCER and BPCER for the KEVR features are shown in Table 2 and Table 3 for the LBG features. Table 4 represents HTER Comparison for State of the Art Methods (SOTAs).

V. DISCUSSION AND ANALYSIS

KEVR Result Analysis:

The analysis of classification results for the features obtained using the KEVR algorithm indicates that for maximum codebook sizes, the Linear SVM and Linear SVM with PCA classifiers have higher accuracy, which is 100% for most of the codebook sizes. The same is true for Logistic Regression with shows higher classification accuracy of 100%. The analysis clearly indicates that with an increase in codebook size, the performance of the classifiers improves, as seen in Fig. 4. The KNN and WKNN perform well for the KEVR algorithm, with accuracy in the range of 99 to 100% for a larger codebook size.

LBG Result Analysis:

The analysis of classification results shows that for the LBG algorithm, the KNN and WKNN classifiers have an accuracy of 100% for larger codebook sizes. The analysis clearly indicates that with an increase in codebook size, the performance of the classifiers is decreased except for KNN and WKNN algorithms, as seen in Fig 5. Logistic regression performs well for smaller codebook sizes of 8 and 16, with 98.51% and 95.77% accuracy.

APCER and BPCER Analysis:

The APCER and BPCER tables clearly indicate that SVM and Logistic Regression classifiers perform extremely well for KEVR codebook features of all the sizes and some variations with satisfactory performance for the KNN classifier and its variant. The BPCER value obtained for the SVM Standardize classifier is 0% for all the Codebook sizes from Codebook size 8 to 256, as shown in Table 2. The Logistic Regression classifier also yields a BPCER value of 0% for all the codebook sizes. In terms of APCER value for KEVR codebooks, Codebook sizes 128 and 256 have values of 0% for all the classification algorithms. In the case of LBG, APCER and BPCER values indicate the dominance of the KNN and WKNN algorithm for all the codebook sizes. BPCER is 0% for KNN and WKNN for all the codebook sizes, and APCER is 0% for larger codebook sizes from Codebook size 16 to 256, as can be seen in Table 3.

HTER Analysis with SOTA:

The results for the HTER parameter for LBG and KEVR algorithms are obtained based on the FPR and FNR for different ML algorithms. The best value for the HTER using the KEVR algorithm is 0 for larger codebook sizes using Logistic Regression. In other terms, the misclassification is 0 for a larger Codebook size using the KEVR algorithm. The LBG algorithm also exhibits an HTER value of 0 for the KNN classification algorithm for larger codebook sizes. Comparison with the relevant methods from literature with reference to HTER is obtained for the 3D MAD Dataset is shown in Table 4. The comparison indicates that the concept of vector quantization for feature extraction and classification using different ML algorithms yields the best

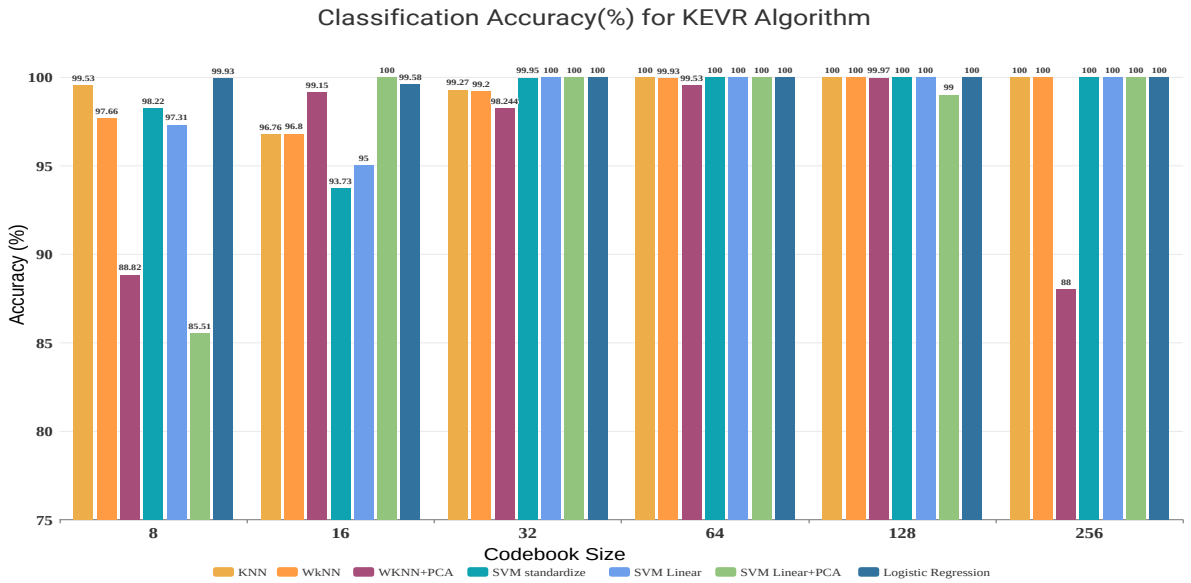

Figure 4. Classification Accuracy for KEVR codebook for Codebook size 8 to 256

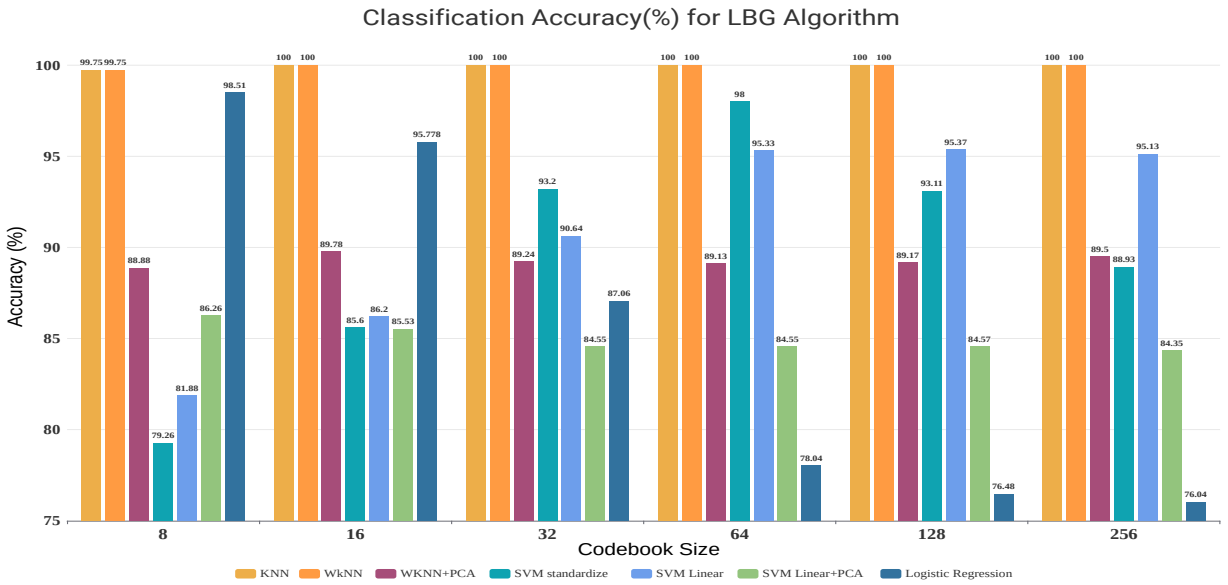

Figure 5. Classification Accuracy for LBG codebook for Codebook size 8 to 256

results for HTER with a value of 0 compared to other techniques proposed in the literature. The technique by Mahore Tripathi, 2018 yields HTER of 0.01% which is the best result in literature on 3D MAD dataset. The proposed model bypasses the best state of art method and yields HTER value of 0%.

VI. CONCLUSION AND FUTURE SCOPE

The widely used biometric authentication method is the Face recognition system, the attacks designed to break the biometrics are spoofing attacks; for face biometric, it's termed as presentation attacks. The 3D mask attacks are performed using masks made of different materials, and these attacks are successful to a greater extent. The proposed system is the first attempt to apply vector quantization algorithms for codebook generation for face presentation attack detection—two primary vector quantization algorithms. LBG and KEVR are used in the proposed system with the first trial of converting the 2D feature vector to 1D form. The codebooks of size variation from 8 to 256 are generated,

Table 2. Table for BPCER and APCER (%) values for KEVR codebooks

Codebook Size/ Algorithm	8		16		32		64		128		256	
	BP*	AP*	BP*	AP*	BP*	AP*	BP*	AP*	BP*	AP*	BP*	AP*
KNN	1	1.41	2	9.74	0.74	2.2	0	0	0	0	0	0
WKNN	2.08	2.8	0	9.6	0	2.4	0	0.21	0	0	0	0
WKNN+PCA	8	14.7	0	2.55	0	5.27	0	1.41	0.09	0	0	36
SVM standardize	0	3.6	0	18.8	0	0.15	0	0	0	0	0	0
SVM Linear	1.8	5.66	0	15	0	0	0	0	0	0	7.27	0
SVM Linear + PCA	3.76	17.5	0	0	0	0	0	0	0	3	14.27	0
Logistic Regression	0	0.21	0	1.26	0	0	0	0	0	0	0	0

Note: BP* - BPCER , AP* - APCER

Table 3. Table for BPCER and APCER (%) values for LBG codebooks

Codebook Size/ Algorithm	8		16		32		64		128		256	
	BP*	AP*	BP*	AP*	BP*	AP*	BP*	AP*	BP*	AP*	BP*	AP*
KNN	0	0.74	0	0	0	0	0	0	0	0	0	0
WKNN	0	0.74	0	0	0	0	0	0	0	0	0	0
WKNN+PCA	0	33.6	0	30.67	0	32.27	10	32.21	10	32.22	0	32.53
SVM standardize	10.37	41.47	10.37	23.07	7.87	4.67	0	2.37	10	6.67	14.94	3.33
SVM Linear	10	34.33	10	21.4	10	8.07	10	8.1	10	9.07	7.27	0.07
SVM Linear + PCA	10	21.2	10	23.4	10	26.33	10	25.67	10	26.33	14.27	19
Logistic Regression	0.37	3.74	6.23	0.2	19.33	0.14	24.94	12.94	23.9	12.34	29	13.33

Note: BP* - BPCER , AP* - APCER

Table 4. Comparison of Proposed system with the Relevant existing method in terms of HTER(%)

Technique	HTER(%)
DWT+LBP(Block 16x16)(24,3) [27]	0.01
DWT+LBP(Block 16x16)(16,2) [27]	0.02
Joint Discriminative Learning [28]	1.76
MS_LBP [3]	12.29
Local and Global features fusion [14]	0.03
Proposed(KEVR+LR)	0
Proposed(LBG+KNN and WNN)	0

which act as the feature vectors. The accuracy obtained for KEVR is 100% for different Codebook sizes for SVM and Logistic Regression Algorithm, and the same is true for KNN and WKNN using LBG. The HTER achieved by the proposed system is 0% which is the best value compared to existing methods. 3D attack detection can be useful to detect attacks on face recognition systems that consider depth information for the detection process, this can be useful in real-time applications that use face images as an authentication method. The proposed system is tested on a 3D MAD dataset, experimentation on 2D datasets and generalization for detection would be the future work.

CONFLICT OF INTEREST

The authors have no conflict of interest.

DATA AVAILABILITY STATEMENT

The Database used in the research is freely available for research work after completing EULA(End user license agreement) formalities. The Data access is made available if EULA is found as per the norms of the data provider. All the permissions and formalities for data access and usage have been done by the authors.

References

[1] S. Bhattacharjee and S. Marcel, "What you can't see can help you - extended- range imaging for 3d-mask presentation attack detection," in 2017 International Conference of the Biometrics Special Interest Group (BIOSIG). IEEE, 2017, pp. 1–7.

[2] D. Wen, H. Han, and A. K. Jain, "Face spoof detection with image distortion analysis," IEEE Transactions on Information Forensics and Security, vol. 10, no. 4, p. 746—761, 2015.

[3] N. Erdogmus and S. Marcel, "Spoofing face recognition with 3d masks," IEEE Transactions on Information Forensics and Security, vol. 9, no. 7, pp. 1084–1097, jul 2014.

[4] T. Edmunds and A. Caplier, "Motion-based countermeasure against photo and video spoofing attacks in face recognition," Journal of Visual Communication and Image Representation, vol. 50, no. 1, pp. 314–332, jan 2018.

[5] N. Daniel and A. Anitha, "Texture and quality analysis for face spoofing detection," Computers Electrical Engineering, vol. 94, no. 6, p. 107293, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790621002731

[6] D. Menotti, G. Chiachia, A. Pinto, W. R. Schwartz, H. Pedrini, A. X. Falcao, and A. Rocha, "Deep representations for iris, face, and fingerprint spoofing detection," IEEE Transactions on Information Forensics and Security, vol. 10, no. 4, pp. 864– 879, apr 2015.

[7] Z. Boulkenafet, J. Komulainen, and A. Hadid, "On the generalization of color texture-based face anti-spoofing," Image and Vision Computing, vol. 77, no. 9, pp. 1–9, 2018.

[8] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face spoofing detection using colour texture analysis," IEEE Transactions on Information Forensics and Security, vol. 11, no. 8, pp. 1818–1830, aug 2016.

[9] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face antispoofing database with diverse attacks," in 2012 5th IAPR international conference on Biometrics (ICB). IEEE, 2012, pp. 26–31.

[10] N. Erdogmus and S. Marcel, "Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect," in 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS). IEEE, 2013, pp. 1–6.

[11] S. P. S. H. B. Kekre, S. D. Thepade and S. Shinde, "Devnagari handwritten character recognition using lbg vector quantization with gradient masks," in 2013 International Conference on Advances in Technology and Engineering (ICATE), 2013, pp. 1–4.

[12] B. Mirzaei, H. Nezamabadi-pour, and D. Abbasi-moghadam, "An effective codebook initialization technique for lbg algorithm using subtractive clustering," in 2014 Iranian Conference on Intelligent Systems (ICIS), 2014, pp. 1–5.

[13] S. T. K. T. S. D. Kekre, H. B. and S. Sanas, "Image retrieval using texture features extracted as vector quantization codebooks generated using lbg and kekre error vector rotation algorithm," in Technology Systems and Management, K. Shah, V. R. Lakshmi Gorty, and A. Phirke, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 207–213.

[14] R. Raghavendra and C. Busch, "Novel presentation attack detection algorithm for face recognition system: Application to 3d face mask attack," in 2014 IEEE International Conference on Image Processing (ICIP). IEEE, oct 2014, pp. 323–327.

[15] A. Pinto, H. Pedrini, W. R. Schwartz, and A. Rocha, "Face spoofing detection through visual codebooks of spectral temporal cubes," IEEE Transactions on Image Processing, vol. 24, no. 12, pp. 4726–4740, 2015.

[16] R. S. F. S Naveen and R. S. Moni, "Face recognition and authentication using lbp and bsif mask detection and elimination," in 2016 International Conference on Communication Systems and Networks (ComNet). IEEE, 2016, pp. 103–106.

[17] P. P. Chan, W. Liu, D. Chen, D. S. Yeung, F. Zhang, X. Wang, and C.-C. Hsu, "Face liveness detection using a flash against 2d spoofing attack," IEEE Transactions on Information Forensics and Security, vol. 13, no. 2, pp. 521–534, 2017.

[18] Y. Ma, L. Wu, Z. Li et al., "A novel face presentation attack detection scheme based on multi-regional convolutional neural networks," Pattern Recognition Letters, vol. 131, no. 3, pp. 261–267, 2020.

[19] D. Mu and T. Li, "Face anti-spoofing with multi-color double-stream cnn," in Proceedings of the 13th International Conference on Distributed Smart Cameras, 2019, pp. 1–4.

[20] S. Kumar, S. Rani, A. Jain, C. Verma, M. S. Raboaca, Z. Illes, and B. C. Neagu, "Face spoofing, age, gender and facial expression recognition using advance neural network architecture-based biometric system," Sensors, vol. 22, no. 14, p. 5160, 2022.

[21] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face antispoofing database with diverse attacks," in 2012 5th IAPR International Conference on Biometrics (ICB), 2012, pp. 26–31.

[22] Y. H. M. B. V. K. Babita Sonare, Shambhavi Mokadam, "Face liveness detection using deep learning and support vector machine," International Journal of Advanced Science and Technology, vol. 29, no. 12, pp. 2566–2572, 2020. [Online]. Available: http://sersc.org/Journals/index.php/IJAST/article/view/24737

[23] G. D. Simanjuntak, K. Nur Ramadhani, and A. Arifianto, "Face spoofing detection using color distortion features and principal component analysis," in 2019 7th International Conference on Information and Communication Technology (ICoICT), 2019, pp. 1–5.

[24] S. Hemajothi, S. Abirami, and S. Aishwarya, "A novel colour texture based face spoofing detection using machine learning," Acta Technica Corviniensis-Bulletin of Engineering, vol. 13, no. 2, pp. 47–52, 2020.

[25] T. de Freitas Pereira, J. Komulainen, A. Anjos, J. M. De Martino, A. Hadid, M. Pietikainen, and S. Marcel, "Face liveness detection using dynamic texture," EURASIP Journal on Image and Video Processing, vol. 2014, no. 1, pp. 1–15, 2014.

[26] I. Chingovska, A. Anjos, and S. Marcel, Anti-spoofing: Evaluation Methodologies, Boston, MA, 2009, pp. 1–6.

[27] A. Mahore and M. Tripathi, "Detection of 3d mask in 2d face recognition system using dwt and lbp," in 2018 IEEE 3rd International Conference on Communication and Information Systems (ICCIS). IEEE, 2018, pp. 18–22.

[28] 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, 2018.

SWAPNIL R. SHINDE Swapnil R. Shinde is currently working as Business Consultant in Convergytics Solutions Pvt. Ltd since January 2022, prior to this role he has Teaching Experience of 12 years with Research Experience of 10 years in the field of Image and Video Processing, Data Security, Biometrics, and Biometric Liveness Detection. He is a member of the Indian Society of Technical Education(ISTE). He has published more than 20 Papers in International/National Conference and Journals.

SUDEEP D. THEPADE Dr.Sudeep D. Thepade is currently Professor in Computer Engineering Department at Pimpri Chinchwad College of Engineering affiliated to Savitribai Phule Pune University, Pune, Maharashtra, India. He completed his Ph.D. in 2011. He has more than 350 research papers to his credit published in International/ National Conferences and Journals. His domain of interest is Image Processing, Image Retrieval, Video Analysis, Video Visual Data Summarization, Biometrics and Biometric Liveness Detection. He is member of International Association of Engineers (IAENG) and International Association of Computer Science and Information Technology (IACSIT). He has served as Technical Program Committee member and Reviewer for Several International Conferences and Journals.

DR.ANUPKUMAR M. BONGALE (SENIOR MEMBER, IEEE) received the Ph.D. degree from Visvesvaraya Technological University (VTU), Belgaum, Karnataka, India. He is currently working as an Associate Professor with the Department of Artificial Intelligence and Machine Learning, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Lavale, Pune, Maharashtra, India. He has filed a patent and has published book chapters. He also published several research articles in reputed international journals and conferences. His research interests include wireless sensor networks, machine learning, optimization techniques, and swarm intelligence. He can be contacted at email: anupkumar.bongale@sitpune.edu.in