Journal of Modern Technology and Engineering
Vol.5, No.1, 2020, pp.48-59

EFFECT OF FEATURE SELECTION WITH META-HEURISTIC OPTIMIZATION METHODS ON FACE SPOOFING DETECTION

Asuman Günay Yılmaz[1*], **Uğur Turhal**[2], **Vasif V. Nabiyev**[1]

[1]Karadeniz Technical University, Trabzon, Turkey
[2]Bayburt University, Bayburt, Turkey

Abstract. Biometric recognition systems are particularly important in real-time systems due to the great convenience they provide to users. Today many mobile phone manufacturers integrate these technologies into their devices with different artificial intelligence based applications and serve them to end-users. However, these systems bring security gaps along with the facilities they offer. Increased use of social media enables the rich content of user information to be used by attackers. Furthermore, visuals and materials, including facial images, bring out high-security risks, especially for face recognition systems. In this study, local texture features were extracted from facial images using local binary patterns. Feature vector size reduced by principal component analysis and three different meta-heuristic optimization algorithms (Particle Swarm Optimization, Ant Colony Optimization, and Simulated Annealing). Then face spoofing detection was performed using support vector machines. Experiments on the NUAA face spoofing database show that that meta-heuristic feature selection methods increase the system performance.

Keywords: Face Spoofing Detection, Artificial Intelligence, Principle Components Analysis, Particle Swarm Optimization, Ant Colony Optimization, Simulated Annealing.
Corresponding author: Asuman, Günay Yılmaz, Karadeniz Technical University, Trabzon, Turkey,
e-mail: gunaya@ktu.edu.tr
Received: 16 September 2019; Accepted: 5 March 2020; Published: 30 April 2020.

1 Introduction

The deployment of biometric-enabled person identification systems has been increasing in recent decades due to the reliability of the person's unique characteristics, such as the face, fingerprint, iris, etc. So many researchers have studied new techniques to improve the performance of these biometric recognition systems. These improvements enable the usage of biometric systems in many application areas such as access control, forensics, surveillance, etc. However, the vulnerabilities of these artificial intelligence systems have not been adequately investigated. In this context, developing biometric systems robust to spoofing attacks has been receiving considerable interest in recent years.

A spoofing attack is an attempt to get access to a biometric system illegally by presenting a synthetic forged (fake) version of the original biometric trait. For example, face images or face videos of a person can be obtained from social networks and used to spoof a biometric face system. Plastic surgery of a face and 3D face masks could be the other traits to face biometric systems. Thus, designing robust anti-spoofing systems is an emerging research field to improve the security of existing biometric systems.

Face spoofing attacks can be grouped into two main categories: 2D face spoofing and 3D face spoofing. 2D face spoofing is performed with the help of photographs and videos. It typically includes printed/digital photo attack and replays video attack. On the other hand, 3D face

spoofing involves 3D mask attack and plastic surgery attack. A printed/digital photo attack is the attempt of spoofing a biometric face system by presenting the printed/digital photograph of an authorized person to the camera of the system. In the replay video attack, the video of the authorized person is presented to the camera of the biometric face system by using a digital device (laptop, tablet, mobile phone). In 3D face spoofing attackers make a 3D mask or alter their facial appearance by plastic surgery to spoof a biometric face system. Thus, 2D face spoofing is more straightforward to use and less costly than 3D face spoofing.

The goal of face anti-spoofing/face spoofing detection systems is to defend the biometric face system against illegal accesses. A face spoofing detection system can be formulated as a two-class classification problem that classifies the input face as real or fake. If the input face is classified as a real face, it is passed to the face recognition system. Otherwise, access is denied.

In this study, the effect of feature selection with meta-heuristic optimization methods on the performance of face spoofing detection system was investigated. The general structure of the system can be seen in Figure 1. After identifying and normalizing the facial region in the input images, local binary patterns (Local Binary Patterns-LBP) and regional features were extracted. Feature size has been reduced with Principal Component Analysis (PCA), classification has been made with the support vector machine (SVM), and the real/fake decision has been made for the images. On the other hand, classification was made by selecting the features from the extracted feature vector with the help of three different meta-heuristic optimization methods (Particle Swarm Optimization-PSO, Ant Colony Optimization-ACO, Simulated Annealing-SA). This paper mainly contributes to exploiting the usage of meta-heuristic optimization methods for feature selection in face spoofing detection problem.

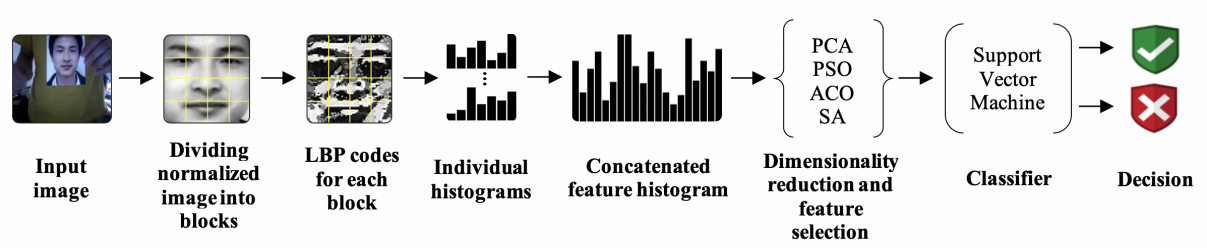

Figure 1: The general structure of the model

The rest of the paper is organized as follows. Some of the face spoofing detection techniques are briefly mentioned in Section 2. The proposed face anti-spoofing method is explained in Section 3. In Section 4, experimental results are given, and the conclusions are outlined in Section 5.

2 Related Works

The existing face spoofing detection systems can be divided into four groups: *i*) texture analysis based methods, *ii*) motion analysis based methods, *iii*) liveness detection based methods, and *iv*) image quality analysis based methods.

Texture analysis based methods use the differences between the texture patterns (print failures, image blur, etc.) of real and fake faces to detect the spoof attacks. These approaches are easy to implement and does not need user collaboration. But they need useful feature vectors to discriminate between real and fake faces. Also, low-quality images or videos which generate low texture information do not give good detection results. Tan et al. (2010) considered the Lambertian reflectance to discriminate between real and fake faces. The method extracts latent reflectance features using a variational retinex-based method and difference-of-Gaussians (DoG) based approach. These features are then used for binary classification. Määttä et al. (2011)

used LBP to analyze the facial texture for face spoofing detection. They applied multi-scale LBP operators (a combination of different LBP operators) to derive enhanced facial representation and capture the differences between real and fake faces. LBP histograms extracted from the whole face and overlapped regions were concatenated and fed to an SVM classifier to make a binary decision about the input image (real or fake). They also extracted texture features using local phase quantization (LPQ) and Gabor wavelets to evaluate the performance of the proposed method. In another study, Määttä et al. (2012), they explored the face spoofing detection performance of both textures based (LBP, Gabor) and gradient-based (Histogram of Gradients-HoG) face descriptors. Face representations generated using these texture descriptors transformed into a compact linear representation by applying a homogenous kernel map on them. Then three SVM classifiers were trained separately with these transformed feature vectors. A final decision (whether the input face is real or fake) is made with a score level fusion of SVM classifiers. Agarwal et al. (2016) proposed a face spoofing detection method using block-wise Haralick texture features (correlation, contrast, entropy, difference variance, sum average, etc.) extracted from redundant discrete wavelet transformed frames of image sequences. The features are extracted from individual color channels of RGB image and concatenated, which yields a high dimensional feature vector. So, dimensionality reduction is performed using principal component analysis, and a two-class SVM classifier is trained to classify the features into spoof and non-spoof classes. Boulkenafet et al. (2016b) proposed a face anti-spoofing method using color texture analysis. They have used different descriptors (LBP, LPQ, BSIF, and SID) to compute texture features from luminance and chrominance channels of different color spaces (HSV and YCbCr). They also fused the different texture features extracted from different color spaces to increase the face spoof detection performance. They have used a linear SVM for binary classification. In another study, Boulkenafet et al. (2016a), proposed a face spoofing detection scheme based on color Speeded Up Robust Features (SURF) and Fisher Vector encoding. First, the SURF features are extracted from different channels of color spaces (HSV, YCbCr) and concatenated to form a feature vector. Then PCA is applied to decorrelate the obtained facial representation and reduce dimensionality. Finally, Fisher vector encoding is used to embed feature vectors into a high-dimensional space more amenable to linear classification. In the classification phase, a Softmax classifier is used to make a decision between real and fake faces. Beham & Roomi (2018) proposed an anti-spoofing enabled face recognition method based on gradient features. Their approach aims to detect spoof attacks by extracting the gradient texture information using local gradient space and weighted gradient-oriented features from the depth map. The classification of real and fake faces is performed using a sparse representation-based classifier (SRC). Zhao et al. (2017) proposed a new a local Spatio-temporal descriptor, volume local binary count (VLBC), to represent the dynamic features. In a local volume, P equally spaced pixels on a circle of radius R in the prior, central, and posterior frames, plus two central pixels in both the prior and posterior frames, are sampled. Then, the sampled pixels are thresholded by the volume center pixel. Furthermore, they build a completed version of VLBC, which includes information about the local difference and the central pixel intensity. They used simple negative log-likelihood distance-based nearest neighbor classifier to evaluate the discriminative capacity of the proposed feature descriptor. Arashloo & Kittler (2018) have proposed a new face spoofing detection formulation based on an anomaly detection concept. In this approach, the training data only taken from the positive class while the test data comes from both positive and negative classes. They have extracted dynamic features from video sequences using different texture descriptors (LBP-TOP, LPQ-TOP, and BSIF-TOP) and image quality measures. They have constructed 20 different two-class (SVM classifier, LDA classifier, and SRC) and one-class (SVM classifier and SRC) classification systems on video sequences to investigate the superiority of one-class anomaly detection approach. Zhang et al. (2018) have proposed a Color Texture Markov Feature (CTMF) and SVM-Recursive Feature Elimination (RFE) based system for face spoof detection. With the help of the directional difference filter, the difference

between real and fake facial texture was determined, and this difference was modeled with the Markov process. With SVM-RFE, the attribute size has been reduced, and the classification process has been performed. Yu et al. (2019) presented a diffusion-based kernel matrix model for facial liveness detecting. They used anisotropic diffusion to improve the edge information of all frames in a video, and also proposed a core matrix model to obtain video attributes that they call diffusion cores. They combined Diffusion Kernel (DK) features with deep attributes from CNN to improve the results.

Motion analysis-based methods are independent of texture and mainly depend on optical flow calculated from video sequences. These methods are difficult to spoof and need low user collaboration. But the need for several video sequences with high motion activity, and high computational complexity are the main drawbacks of these approaches. Anjos et al. (2013) proposed a counter-measure based on foreground/background motion correlation using optical flow. They first compute the direction of motion for every pixel using the horizontal and vertical orientations. Then the normalized histograms for face and background regions were calculated, and X^2 distance between the angle histograms of face and background regions was computed. Finally, the X^2 scores were averaged over a window size of N frames, with a possible specified overlap size. They used a binary classifier to distinguish between real-accesses and spoofing attempts.

Liveness detection-based techniques try to detect physiological signs of life in videos such as eye blinks, facial expressions. But these methods need high user cooperation, other devices, and video sequences. Also, they are time-consuming and computationally complex. Alotaibi & Mahmood (2017) proposed a face liveness detection method that uses nonlinear diffusion to obtain depth information and preserve boundary locations. Then a deep convolution neural network that leads a better classification is used to extract the discriminative and high-level features from diffused images.

As the image quality properties of real accesses and illegal attacks are different, image quality analysis-based methods compare the image qualities of real and fake faces like color diversity, blurriness, edge information, chromatic moment features, etc. The main advantages of these methods are they are easy to implement, have a low computational cost, and no user collaboration is needed. But their performance is highly dependent on the quality of the images. Galbally et al. (2013) have implemented 25 image quality features (mean square error, peak signal to noise ratio, maximum difference, average difference, etc.) to distinguish between genuine and spoof faces. They have used Linear Discriminant Analysis and Quadratic Discriminant Analysis for classification. Wen et al. (2015) proposed an image distortion analysis (IDA) based face anti-spoofing algorithm. They extracted four different IDA features (specular reflection, blurriness, chromatic moment, and color diversity) from face images and concatenated to form a feature vector. The feature vector is fed into an ensemble of SVM classifiers, each trained on a different group of training data. Then a binary decision (real or fake) is made by fusing these classifier's outputs. They also proposed a multi-frame fusion scheme to detect the spoofing attacks on videos. For an image sequence, the classification results are obtained from individual frames and combined using a voting scheme to get a spoof detection score.

3 Proposed Method

The proposed face spoofing detection method consists of image normalization, feature extraction, dimensionality reduction and feature selection, and classification modules. These modules are explained in detail in the following subsections.

A. Image Normalization

The input images are 640 x 480 pixels in size and contain unnecessary features such as background, cloth, and hair. So, the image normalization step is performed to extract

only the facial regions and to adjust the orientation and size of the faces. First, the facial area was detected from the images. The resulting facial images were aligned according to eye center positions and scaled to 64 x 64 pixels. Finally, the images were converted to 8-bit gray-level images.

B. Feature Extraction with LBP

LBP is a robust method that defines texture information independent from gray levels in digital images (Ojala et al., 2002). The original LBP operator compares each pixel to the pixels in its 3 x 3 neighborhood region and ultimately generates a binary code for that center pixel. The LBP codes are generated by (1).

$$LBP_{P,R}(x_c) = \sum_{p=0}^{P-1} u(x_p - x_c)^{2p} \tag{1}$$

$$u(y) = \begin{cases} 1 & \text{if } y \geq 0 \\ 0 & \text{if } y < 0 \end{cases}$$

In (1), x_c is the center pixel, x_p denotes the neighbors of the center pixel, R, refers to the distance between the center pixel and its neighbors and P is the number of neighbors. Not all the LBP codes are used in texture description. The uniform patterns which contain at most two bitwise transitions from 0 to 1 or vice versa are used in texture recognition. The LBP histogram is produced on the input image $I_{(x,y)}$ for $P = 8$ neighborhood at a distance of $R = 1$ pixel using (2) (Günay & Nabiyev, 2017).

$$H = \sum_{x_c \in I_{x,y}} f\{LBP_{8,1}(x_c) = U(i)\} \tag{2}$$

$$f(y) = \begin{cases} 1, & \text{if } y \text{ is true} \\ 0, & \text{if } y \text{ is false} \end{cases}$$

$U(i)$, is the sequence that holds 58 uniform patterns produced in 8 neighborhoods, and H is the LBP histogram.

C. Dimensionality Reduction and Feature Selection

In this section, the PCA used in size reduction and meta-heuristic optimization methods (PSO, ACO, SA) used in feature selection are explained.

i. *Principal Component Analysis:* PCA is a subspace projection method which is widely used in pattern recognition studies. PCA is a standard tool commonly used to reduce the size of multivariate data. The technique looks for linear combinations of variables called principal components that best represent the dataset. Principal components correspond to eigenvectors that maximize the variance of the data projected onto them (Croux et al., 2013). The eigenvectors of the data covariance matrix (S) are obtained by equation $W_{opt} = argmax_{||W||=1} W^T S W$. When the equation is solved, the eigenvectors (W) corresponding to the largest $d(d \leq D)$ eigenvalue of S are obtained. Then dimensionality reduction is performed with $y_i = W^T x_i (y_i \in R^d)$ (Günay & Nabiyev, 2017). In the study, principal components with 95% eigenvalues were used.

ii. *Particle Swarm Optimization:* PSO optimization algorithm developed in 1995 on the discovery that animal clusters that move in the herd are engaged in inter-individual interaction in the process of eliminating their basic needs, such as foraging and that

this interaction affects the process of achieving the goal (Kennedy & Eberhart, 1995). In this algorithm, each individual searching in the process of reaching the solution is called a particle, and the population of the particles is called a swarm. The fitness function is used to figure out how close the particles are to the solution. The algorithm steps of particle swarm optimization are given in Algorithm 1 (Brownlee, 2011).

Algorithm 1: Particle Swarm Optimization Psudocode

```
Input  : ProblemSize, Population_size
Output: P_g_best
Population ← ∅
P_g_best ← ∅
for i = 1 to Population_size do
    P_velocity ← RandomVelocity()
    P_position ← RandomPosition(Population_size)
    P_p_best ← P_position
    P_i ∈ Population
    if (Cost(P_p_best) ≤ Cost(P_g_best)) then
        P_g_best ← P_p_best
while (¬StopCondition()) do
    for (P ∈ Population) do
        P_velocity ← UpdateVelocity(P_velocity, P_g_best, P_p_best)
        P_position ← UpdatePosition(P_position, P_velocity)
        if (Cost(P_position) ≤ Cost(P_p_best)) then
            P_p_best ← P_position
            if (Cost(P_p_best) ≤ Cost(P_g_best) then
                P_g_best ← P_p_best
Return: P_g_best
```

In Algorithm 1, $UpdateVelocity$ is used to update the accelerations of the population individuals to reach the target by using the best solution values. $UpdatePosition$ is used to update the positions of the individuals using their momentum and the best position values.

iii. *Ant Colony Optimization:* This algorithm was developed as a result of discovering the behaviors of some ant species in the process of searching for food sources. In this process, ants secrete a fragrance called pheromone and thus find the shortest path between their nests and the source. Similar to real ants, ACO has agents called artificial ants. The pheromone amount of each artificial ant deposit is proportional to the quality of the solution created by the artificial ant (Dorigo & Birattari, 2010). In the algorithm, all individuals looking for a solution are called ants, and the population formed by individuals is called a colony. The algorithm steps of ant colony optimization are given in Algorithm 2 (Brownlee, 2011).

iv. *Simulated Annealing:* The algorithm is based on the similarity relationship between the annealing of solids and the complexity of resolution of major optimization problems. Annealing is the process of heating the solid to its melting point and then cooling it slowly until it crystallizes. The atoms of the materials are at high energy levels at high temperatures have greater freedom of movement for smooth placement. The system has minimum energy when a properly structured crystal is provided. As the temperature decreases, atomic energy decreases. If the cooling process takes place too quickly, defects, and irregularities in the crystal structure will occur. For this reason, cooling should be done carefully.

According to the similarity between the combinatorial optimization problem and the annealing process, the states of the solid represent possible solutions of the optimiza-

tion problem, and their energies correspond to the purpose function values calculated for the solutions. The minimum energy state represents the optimal solution for the problem. SA is an iterative algorithm; that is, the algorithm continually tries to develop a solution expressed in the form of a vector of numbers in the solution space (Kalinli & Karaboga, 2004). The algorithm steps of simulated annealing are given in Algorithm 3 (Brownlee, 2011).

Algorithm 2: Ant Colony Optimization Psudocode

Input : ProblemSize,$Population_{size}, m, \rho, \beta, \sigma, q^0$
Output: P_{best}
$P_{best} \leftarrow$ CreateHeuristicSolution(ProblemSize)
$Pbest_{cost} \leftarrow$ Cost(S_h)
$Pheromone_{init} \leftarrow$ [1.0 / (ProblemSize x $Pbest_{cost}$)]
Pheromone $\leftarrow$ InitializePheromone($Pheromone_{init}$)
while *(¬StopCondition())* **do**
  for i = *1* **to** m **do**
    $S_i \leftarrow$ ConstructSolution(Pheromone,ProblemSize,β, q^0)
    $Si_{cost} \leftarrow$ Cost(S_i)
    if *($Si_{cost} \leq Pbest_{cost}$)* **then**
      $Pbest_{cost} \leftarrow Si_{cost}$
      $P_{best} \leftarrow S_i$
    LocalUpdateAnyDecayPheromone(Pheromone,S_i, Si_{cost}, σ)
  GlobalUpdateAnyDecayPheromone(Pheromone,$P_{best}, Pbest_{cost}, \rho$)
Return: P_{best}

Algorithm 3: Simulated Annealing Algorithm Psudocode

Input : ProblemSize,$iterations_{max}, temp_{max}$
Output: S_{best}
$S_{current} \leftarrow$ CreateInitialSolution(ProblemSize)
$S_{best} \leftarrow S_{current}$
for i = *1* **to** $iterations_{max}$ **do**
  $S_i \leftarrow$ CreateNeighborSolution($S_{current}$)
  $temp_{current} \leftarrow$ CalculateTemperature($i, temp_{max}$)
  if *(Cost(S_i) $\leq$ Cost($S_{current}$))* **then**
    $S_{current} \leftarrow S_i$
    if *(Cost(S_i) $\leq$ Cost(S_{best}))* **then**
      $S_{best} \leftarrow S_i$
  else if (exp[($CostS_{current} - CostS_i$)/$temp_{current}$] > $Rand()$) **then**
    $S_{current} \leftarrow S_i$
Return: S_{best}

The feature selection process, with the help of meta-heuristic optimization methods, is carried out by optimizing the average error value obtained from artificial neural network (ANN), which is used as a fitness function. First, random starting positions are assigned to each individual contained in the populations. The dataset consisting of n features (depends on PCA) with the lowest position value is sent to the fitness function, and the average error value is used as the cost value. These procedures are repeated for each iteration. When updating the positions of individuals, the previous best position of each individual and the best position values of the population set is used. At the end of all iterations, the features which produce the lowest cost value in all populations are selected. In the study, the LBP features that are more effective for face spoofing detection were selected, as explained using optimization methods.

In optimization algorithms, iteration and population selection are an optimization problem in itself. While high-population solutions can produce more precise results, in some cases, they may not contribute to the problem. Choosing the iteration and population sizes are dependent on the size of the problem set. In small datasets, these values may not affect the calculation times. However, the fact that the data set used in this study contains a high number of features causes high computational complexity and memory consumption.

In this study, 20 iterations and 50 populations were selected for use in each iteration. As a result of the experimental studies for the data set used, it was seen that the optimization algorithms repeat similar results after 20 iterations. Population size is an optional parameter, determined according to the limits of the computer where the experiments are carried out. Besides, the average error value was obtained from the classification of the ANN with 70% education, 15% verification, and 15% test parameters.

D. Classification

Support Vector Machines (SVM) is a controlled classification algorithm based on statistical learning theory. The mathematical algorithms owned by SVM were initially designed for the problem of classification of two-class linear data, and then generalized for classification of multi-class and non-linear data. SVM is based on the definition of the hyperplane that can optimally separate the two classes from each other (Kavzoğlu & Çölkesen, 2010). An example model for separating the two classes is given in Figure 2.

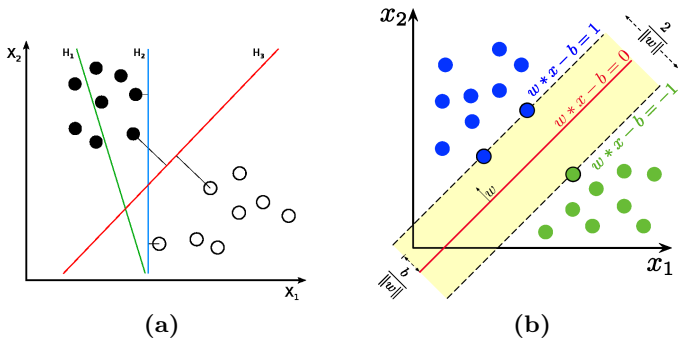

Figure 2: Detection of hyperplane separating the two classes

In Figure 2(a), H1 does not separate the classes. H2 does, but only with a small margin. H3 separates them with the maximal margin. In Figure 2.(b), the plane passing between the closest examples of the two classes is the optimum point in the separation of these two classes. In the determination of this plane, samples that touch virtual points that are equidistant to the plane are called support vectors. In the study, SVM with Radial Basis Function (RBF) core was used to classify the input image as real/fake.

4 Experimental Results

In this section, the results obtained using the database, performance criteria, and proposed approach used in the experiments are explained in detail.

NUAA PID (Tan et al., 2010) is a publicly available database that consists of both real access and spoof attack attempts of 15 subjects. The videos were captured under uncontrolled illumination conditions in three different sessions with two-week intervals. Details of the data set are given in Table 1.

Table 1: Summary of NUAA face spoofing database

# of subjects	# of images/ videos	Attack type	Illumination condition
15	5015 genuine images 7509 spoof images	Printed photo	Uncontrolled

In real access attempts, no face movements are permitted. The attacks were conducted by printing the user photographs professionally on photographic papers of sizes 6.8cm x 12cm, 8.9cm x 12.7cm, and on A4 70gr paper using a standard color ink HP printer. The printed images were moved back and forth vertically and horizontally and slightly warped around during the attacks to imitate the real accesses. The training set comprises images from the first two sessions while the test set contains the samples from the last session. Examples of attacks are given in Figure 3.

Figure 3: Sample attack images of NUAA database

The NUUA database used in the experiments consists only of training and test sets. For this reason, the training set was separated as a training and development set using 5-fold cross-validation, and the average of all the results obtained was given as system performance. The design of the classification model is given in Figure 4.

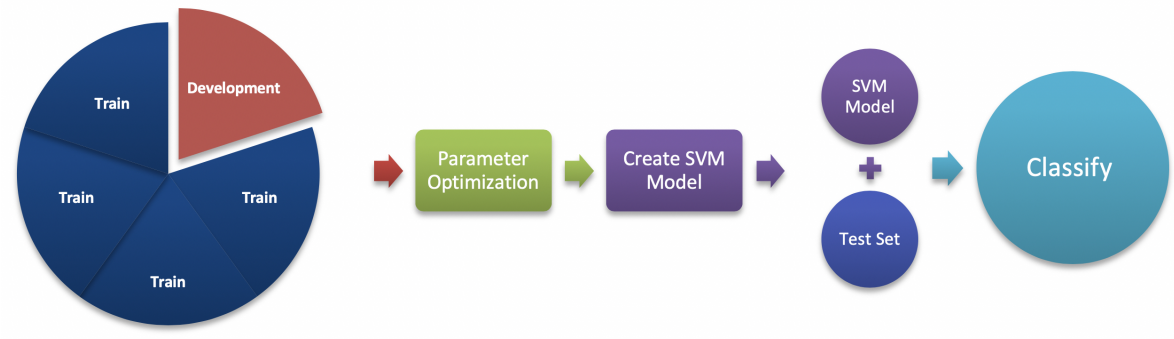

Figure 4: Design of the classification model

A face spoofing detection system faces two types of errors. These are true access denial (false rejection) and acceptance of an attack (false acceptance). The performance of these systems is usually measured by half total error rate (HTER) metric. HTER is the half of the sum of False Acceptance Rate-FAR and False Rejection Rate-FRR (3).

$$HTER(\tau) = \frac{FAR(\tau) + FRR(\tau)}{2} \tag{3}$$

Since FAR and FRR depend on the threshold value of τ, increasing FAR causes the FRR to decrease. The opposite is also true. For this reason, the results are usually represented by ROC curves showing the change of FAR relative to FRR for different τ threshold values. Another criterion used for face spoofing detection is the equal error rate (EER). EER is the value at the point where the FAR equals FRR on the ROC curve. The threshold value corresponding to this value is obtained from the development set, and HTER is reported on the test set using this threshold value.

In the study, EER, HTER, Precision, and classification time criteria were used to evaluate the system performance. Precision (4) is obtained from the confusion matrix, which is used to evaluate the results of classification algorithms.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

In the equation, TP represents the number of correctly classified positive samples, while FP represents the number of false classified positive samples.

All classification processes were carried out using a laptop with an Intel i7-2.4GHz processor, 8GB RAM, and 64-bit operating system. Calculation times refer to the time elapsed during the classifier parameter optimization, entire classifier model generation, and estimation process.

For an efficient representation and also consider shape information, local LBP histograms can be used. For this purpose, the image can be divided into regions, and for every region, local histograms are produced. Then these regional histograms are concatenated to build a global description of the image. In the study, the images were LBP histograms are extracted from the whole image, and face spoofing detection is performed using these texture descriptors. Then the images are divided into a various number of discrete blocks (32x32, 16x16, 8x8 pixel sub-regions). Spatial LBP histograms are extracted from these blocks and concatenated to form a spatial texture descriptor. When the face spoofing detection performance of these features is calculated, it seems that spatial features extracted using 16x16 sub-regions give the best performance. For this reason, all the images were divided into 16 x 16 pixel sub-regions for feature extraction in the study.

Each input image was divided into 16 x 16 pixel sub-regions, and $LBP_{8,1}$ operator is used to extract LBP histograms from these sub-regions. Then regional histograms were concatenated to form a spatial texture descriptor. After dimensionality reduction with PCA, face spoofing detection is performed. On the other hand, classification performance was examined by performing the feature selection process from the regional LBP histograms, using PSO, ACO, and SA meta-heuristic optimization algorithms. The number of features selected in the optimization algorithms was determined by the number of features obtained as a result of dimensionality reduction with PCA. The classification performance of the proposed face spoofing detection system is given in Table 2. It can be seen from the table that the features selected with the SA algorithm produced the lowest HTER value, with 12.18% and the highest precision value with 83.54%. However, 15.66% HTER and 72.44% precision, and 18.78% HTER and 65.10% precision values were obtained using the features selected with ACO and PSO algorithms, respectively. The PCA algorithm, which is frequently used in the literature for dimensionality reduction, produced the lowest results after PSO with 18.00% HTER and 66.45% precision.

When the classification times examined, it was found that the data set that was not subjected to the feature selection process was classified approximately five times longer than the data sets that were subjected to the dimensionality reduction and feature selection process. This reveals the positive effect of dimensionality reduction and feature selection algorithms on the classification time. However, an increase of 20.51% was observed in the highest precision value obtained after the feature selection process compared to the data set without the feature selection.

Table 2: Classification result of NUAA dataset

Method (# of Features)	**EER (%)**	**HTER (%)**	**Precision (%)**	**Classification Time (sec)**
LBP- (944)	0	17.12	69.31	2766.96
LBP + PCA- (185)	0	18.00	66.45	613.64
LBP + PSO- (185)	0	18.78	65.10	566.06
LBP + ACO- (185)	0	15.66	72.44	581.02
LBP + SA- (185)	0	**12.18**	**83.54**	601.35

5 Conclusions

In this study, the effect of feature selection with meta-heuristic optimization methods on face spoofing detection performance was investigated. The input images were divided into 16 equivalent blocks, and the features were extracted with LBP. The size of the feature set has been reduced by various optimization algorithms such as PSO, ACO, SA. The same process was carried out using the PCA algorithm, which is frequently used in the literature for size reduction. The obtained datasets were classified with SVM-RBF, and performance analysis was performed with various performance parameters.

The results obtained reveal the positive effects of feature selection processes performed with the help of meta-heuristic optimization algorithms on face spoofing detection. Calculation times have decreased in all algorithms, and performance has increased in SA and ACO algorithms. Hybrid models can be useful for improving classification performance.

References

Agarwal, A., Singh, R., & Vatsa, M. (2016, September). Face anti-spoofing using Haralick features. In *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)* (pp. 1-6). IEEE.

Alotaibi, A., Mahmood, A. (2017). Deep face liveness detection based on nonlinear diffusion using convolution neural network. *Signal, Image and Video Processing, 11*(4), 713-720.

Anjos, A., Chakka, M.M., & Marcel, S. (2013). Motion-based counter-measures to photo attacks in face recognition. *IET Biometrics, 3*(3), 147-158.

Arashloo, S.R., Kittler, J. (2018). An anomaly detection approach to face spoofing detection: A new formulation and evaluation protocol. *IEEE International Joint Conference on Biometrics, IJCB*, 80-89.

Beham, M.P., Roomi, S.M.M. (2018). Anti-spoofing enabled face recognition based on aggregated local weighted gradient orientation. *Signal, Image and Video Processing, 12*(3), 531-538.

Boulkenafet, Z., Komulainen, J., Hadid, A. (2016a). Face antispoofing using speeded-up robust features and fisher vector encoding. *IEEE Signal Processing Letters, 24*(2), 141-145.

Boulkenafet, Z., Komulainen, J., & Hadid, A. (2016b). Face spoofing detection using colour texture analysis. *IEEE Transactions on Information Forensics and Security, 11*(8), 1818-1830.

Brownlee, J. (2011). *Clever Algorithms: Nature-Inspired Programming Recipes*. Jason Brownlee.

Croux, C., Filzmoser, P., & Fritz, H. (2013). Robust sparse principal component analysis. Technometrics, 55(2), 202-214.

Dorigo, M., Birattari, M. (2010). *Ant Colony Optimization.* Encyclopedia of machine learning. Springer US.

Galbally, J., Marcel, S., & Fierrez, J. (2013). Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition. *IEEE Transactions on Image Processing, 23*(2), 710-724.

Günay, A., Nabiyev, V. (2017). Determining the age estimation accuracies of facial regions under age groups. *TBV Journal of Computer Science and Engineering, 9*(2), 1-10 (in Turkish).

Kalinli, A., Karaboga, N. (2004). Design of the high order iir filters using simulated annealing algorithm. *Erciyes University Journal of the Institute of Social Sciences, 20*(1-2), 20-27 (in Turkish).

Kavzoğlu, T., Çölkesen, I. (2010). Investigation of the effects of kernel functions in satellite image classification using support vector machines. *Harita,* 144, 73-82.

Kennedy, J., Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks* (Vol. 4, pp. 1942-1948). IEEE.

Määttä, J., Hadid, A., & Pietikäinen, M. (2012). Face spoofing detection from single images using texture and local shape analysis. *IET biometrics, 1*(1), 3-10.

Määttä, J., Hadid, A., & Pietikäinen, M. (2011, October). Face spoofing detection from single images using micro-texture analysis. In *2011 international joint conference on Biometrics (IJCB)* (pp. 1-7). IEEE.

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence, 24*(7), 971-987.

Tan, X., Li, Y., Liu, J., & Jiang, L. (2010, September). Face liveness detection from a single image with sparse low rank bilinear discriminative model. In *European Conference on Computer Vision* (pp. 504-517). Springer, Berlin, Heidelberg.

Wen, D., Han, H., & Jain, A.K. (2015). Face spoof detection with image distortion analysis. *IEEE Transactions on Information Forensics and Security, 10*(4), 746-761.

Yu, C., Yao, C., Pei, M., & Jia, Y. (2019). Diffusion-based kernel matrix model for face liveness detection. *Image and Vision Computing*, 89, 88-94.

Zhang, L.B., Peng, F., Qin, L., & Long, M. (2018). Face spoofing detection based on color texture Markov feature and support vector machine recursive feature elimination. *Journal of Visual Communication and Image Representation,* 51, 56-69.

Zhao, X., Lin, Y., & Heikkilä, J. (2018). Dynamic texture recognition using volume local binary count patterns with an application to 2D face spoofing detection. *IEEE Transactions on Multimedia, 20*(3), 552-566.