A Novel Deep Learning-Based Method for Real-Time Face Spoof Detection

Muhammad Amir Malik
International Islamic University

Tehseen Mazhar (✉ tehseenmazhar719@gmail.com)
Virtual University of Pakistan

Inayatul Haq
Zhengzhou University

Tariq Shahzad
COMSATS University Islamabad

Yazeed Yaseen Ghadi
Al Ain University

Fatma Mallek
University of de Moncton

Habib Hamam
University of Ha'il

Research Article

Keywords: Face Spoof Recognition, Machine Learning, Computer Vision, CNN, Image Processing, Deep learning.

Posted Date: September 29th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3371756/v1

License: © This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License
Abstract

Facial-based Commercial Off-The-Shelf (COTS) systems increase the success rate of spoof attacks to 70% detection accuracy. A spoof attack uses an image, video, or 3D model of a person to gain unauthorized access to a biometric system. Face spoof attacks are mostly based on common spoof vectors, print attacks, and replay attacks. This research aims to improve the detection accuracy of face spoof recognition systems by employing a hybrid model of machine learning and computer vision-based approaches. Differences, including Decision Tree, Naive Bayes, K-nearest Neighbor, Support Vector Machine, Convolutional Neural Network (CCN), and Recurrent Neural Networks are used for face spoof detection. For face spoof detection, the proposed model is a hybrid variant of the CNN-based classifier used in the proposed face spoof detection model. This study improves real-time face fake detection using machine learning and computer vision. The proposed system is based on a CNN-based classification approach with optimized hyper parameters that detect real-time face spoofing attacks using print, video, and repeat attacks, improving detection accuracy. IDIAP, USSA, and MSFD datasets are used in the simulation; the proposed model has achieved a maximum accuracy of 87.5%. Furthermore, the proposed model achieved a high sensitivity score of 92.45%, indicating that it is highly likely to be used for spoof attack detection systems in the future.

1. Introduction

Facial spoofing is stealing someone’s identity by using a photograph or video of their face to trick facial recognition software and steal their identity (becoming a digital identity theft) [1, 2]. This spoofing attack aims to deceive the facial biometric identification control systems and steal someone else’s identity by mimicking or using that person’s face. People do this by using their faces or mimicking the other person’s face. On the other hand, spoofing face recognition has become a reality due to the advancement of anti-spoofing technology and cybersecurity-related solutions. Facial anti-spoofing protection is an anti-spoofing technology that, by definition, detects and restricts the use of print or video-based facial features for stealing the rights of others in various business-related areas like their bank, mobile phone providers, or insurance companies.

Human facial recognition is becoming increasingly crucial as security demands and technological developments rise. Facial recognition systems can be deceived due to how simple it is to gather contactless face data [1]. Criminals utilize spoofing to get around security measures by assuming another person’s identity.

The authors in [2] stated that Deep Learning anti-spoofing techniques are necessary to ensure the security of facial recognition systems in real-time. Attackers have started fooling facial recognition systems by spoofing (also known as ”Presentation attacks” or ”PA”) them. The three main tools used in PA include a mask, a paper copy of the man’s face, and a digital device that replays a video of the person’s face. Presentation attack. (Mask attack). Attacks using fabrication should not be possible against facial recognition systems. Deep learning systems were previously set up as binary classification
problems to avoid faking. Fig 1 shows Examples of images (a) and (b) representing two spoofing attempts where two different subjects (left) apply makeup (middle) to resemble the same target identity (right).

To avoid spoofing, it uses supervised technologies like binary or supplementary control. Most lack generalization and cross-dataset resilience [3]. RGB images and videos are commonly used as inputs for systems that detect fake faces and prevent spoofing [2]. Texture-based methods allow the use of specially crafted features to prevent spoofing. Soft-max loss tests various Convolutional Neural Network algorithms while advancing deep learning [2].

According to [4], deep anti-spoofing models with binary supervision have two main drawbacks. The first thing you’ll notice when contrasting a spoof image with one of a natural person is that patterns, forms, and spoof objects have taken the place of skin tone (e.g., 0 reflections). A Convolutional Neural Network with softmax loss could recognize random signals like those on the screen’s edge. Still, it could not distinguish between spoof patterns and random signals. These models make inaccurate generalizations due to their inability to differentiate between natural and artificial faces. Even worse, models developed with binary supervision merely present a choice between two options without any background information or justification.

Facial spoofing uses a picture, video, disguise, or other things to replace an authorized person’s face. There are numerous options for how to complete this. One example is a print attack. When someone exploits a photograph of another person for this purpose, it is called a print attack. The image can be displayed on a computer device or printed.

Similarly, an attempt to obtain access without authorization by intercepting transmitted identifying or access control information and transmitting it again. One example of this kind of attack is a repeat effort. The Replay-Attack Collection for face spoofing comprises 1,300 video clips of failed photo and video attacks on 50 customers, all shot in different lighting conditions. On the other hand, identifying someone by their face is known as "three-dimensional facial identification" because it uses the face’s three-dimensional form. It has been demonstrated that 3D face recognition methods are more accurate than their 2D equivalents. An RGB-D camera takes numerous video frames of a person’s image while wearing a 3D face mask with varying levels of light, which illustrates a 3D mask attack.

Facial spoofing is stealing someone’s identity by using a photograph or video of their face to trick facial recognition software and steal their identity (becoming a digital identity theft). Face recognition has become a reality due to the advancement of anti-spoofing technology and cybersecurity-related solutions. The problem is to devise a real-time Facial anti-spoofing method that can accurately detect and restrict the use of print or video-based facial features for stealing the rights of others in various business-related areas like their bank, mobile phone providers, or insurance companies. This paper contributes to the literature by offering a more accurate method of face spoof detection for real-time scenarios using deep machine learning and a computer vision-based mixed approach.

The proposed method is evaluated on three datasets to validate the detection result.
- To achieve high detection accuracy and sensitive rate against print, video, and replay attacks for face spoof detection.

The proposed method incorporates an ML classifier with reportedly better classification results than other classifiers. This classifier is selected after developing a module given in Fig. 4 and running it over various datasets. The module given in Fig 4 reported CNN with maximum accuracy after running over various datasets. After that, an optimized CNN architecture (with modified layers and hyperparameters) is selected for maximum accuracy for real-time detection of face spoof attacks.

The data set used in this work is obtained from the Kaggle database, which consists of different images. The dataset consists of three face spoof databases: the MSU MFSD Database of photo and video attack attempts, the IDIAP dataset of Replay attacks, and SiW of videos, including live and spoof videos. These wide-spanned datasets were selected to evaluate the proposed model's effectiveness in face spoof attacks.

The rest of the paper is divided into the following sections. Section 2 discusses the related work. Section 3 discusses the proposed model's materials, methods, and details for detecting face spoof attacks. Section 4 discusses the Results and Simulation. Section 5 reports a discussion on the obtained results. Section 6 discusses future work.

2. Related Works

A real face can be classified using texture, motion, and depth information [4]. The authors have proposed a CNN-based binary supervision method with soft max loss to detect the motion of objects [5]. The authors in [6] presented a model to identify translation, rotation, and swing movements using Markov color texture-based face spoof detection and support vector machine recursive feature reduction. The authors in [7] worked on removing noise to handle image losses due to noises. They stated that factors like local highlights, shadows, blurred photographs, and others add to the illusion and must be handled for face spoof detection.

Authors in [8] used a histogram-based model to detect and classify real and fake faces. The RGB, YCbCr, and HSV color spaces were compared to determine the most accurate color system. The authors [9] used a DOG filter, collected medium-frequency band picture data, identified vital features using Fourier transformation, and then divided and grouped the data using logistic regression. The researchers in [10] developed an interactive anti-spoofing detection algorithm to help avoid compromise systems for spoof recorded videos. The authors [11] developed a method to recognize the face-mouth region change. They used participants’ blinking and facial movements to identify whether the eyes and mouth were open by looking at the surface of the eyes and teeth and their color, saturation, and value. Authors in [12] developed a system that allows humans and computers to work together to help people produce random facial expressions. Users can decide whether the appropriate range of facial emotions was captured by looking at many photos and computing the SIFT flow energy for each one. The authors in [13] used the optical Flow of Lines to calculate the difference in space and time between horizontal and vertical
images of faces to detect attacks using videos and 3D masks. Authors in [14] claimed that convolutional neural network-based deep Learning outperforms other feature extraction techniques. The authors in [15] developed a Face spoof attack detection system with and without masks using infrared short-wave technology. The picture pixels of a depth camera are used to calculate the separation between two objects. The authors in [16, 18, 21] used transfer learning to avoid over-adapting to a big network structure built on VGG16. A study [17] developed a FAS Net network framework that uses a pre-trained Convolutional Neural Network to identify fake faces. Another study [19] used Haar-like features and Linear Discriminant Analysis to analyze faces in photos and detect spoofing attacks using the co-occurrence of adjacent local binary patterns (CoALBP). The author [20] used Fourier spectra of both 2D and 3D images to discover textural differences. The authors in [21] used Difference-of-Gaussian (DoG) filters to extract texture data from beneath the surface of faces to differentiate between real and synthetic faces. The authors in [22,23] used pre-trained architectures with global features for face spoof detection. The author [24] used random patches to classify structure through facial features.

The authors in [25] used a backpropagation algorithm with various features like the shape of eyes, etc., for detecting 3D mask attacks based on identifying the pulse in films, which might differ depending on how the camera is set up and how intense the light is [25]. The authors in [26] give a novel attention-based fusion technique to develop a human visual system model. The authors in [27] used a two-stream convolutional neural network, one for the multi-scale retinex space and the other for the RGB color space, to classify facial features. The author [28] used Conditional Random Fields (CRF) to forecast blinking based on Local Binary patterns and Simplified Weber Local Descriptor encoded Convolutional Neural Network models. The authors in [28] used a Support Vector Machine (SVM) classifier to identify spoofing. A study [29] stated that Fully Convolutional neural networks perform better than Convolutional Neural Network (CNN)-based techniques. A study [33] proposed a Patch Net-based framework that detected patches based on spoof attacks cropped from non-distorted face images. The authors in [34] used a 5-layered U-Net-based soft bio-metric methodology for face detection and Alex Net-based architecture for facial information based on age, gender, facial expression, face spoofing, etc. The authors in [35] used a modified neural network model for face-anti-spoofing that outperformed other networks due to extensive training. The authors in [36] correctly identified test encrypted faces using a PCA-Based Face Recognition System. The authors in [46] proposed a computer vision-based approach for assessing the surface quality of concrete buildings. This method uses convolutional neural networks (CNNs), transfer learning, and decision-level image fusion to assess the condition of a concrete building’s exterior using simple fractures to determine the structural soundness to resolve one of the main issues with current diagnostic technologies, particularly regarding their lengthy processing times, subjectivity, and reliance on the surveyors' prior knowledge. The researchers in [47] proposed an enhanced bird swarm method to optimize the CNN hyperparameters based on a 2D CNN-based data-driven model to figure out the rotational capacity of reinforced concrete (RC) columns taking input as concrete compressive strength, steel ratio of longitudinal & transverse reinforcement, stirrup spacing, and beam height, stirrup width, and stirrup height (CNN) as inputs. From the above literature, it has been shown that the deep anti-spoofing models with binary supervision have two main drawbacks. The first thing you'll notice when contrasting a
spoof image with one of a natural person is that patterns, forms, and spoof objects have taken the place of skin tone (e.g., 0 reflections). A Convolutional Neural Network with softmax loss could recognize random signals like those on the screen's edge. Still, it could not distinguish between spoof patterns and random signals.

3. Materials and Methods

For real-time applications, several researchers have looked at face anti-spoofing techniques. The majority of this work is done manually, mainly feature extraction. The highest-risk tasks remained printing, film rewinding, and making 3D masks. The authors in [4] used the FAS Net Network Framework is based on transfer learning, while [20, 31] uses a hyper-deep translation-based, data-driven technique. Also, [32] used the FAS Net Network Framework based on transfer learning. Machine learning has increasingly concentrated on transfer learning through domain generalization or adaptation. Researchers have looked into LSTM, Auxiliary supervision, a machine learning method based on visual translation, and GAN. This provides an accurate and effective way to prevent face spoofing in real-time applications [45].

The foundations for contrasting deep Learning and machine learning research are also examined. With a learning module that assesses the effectiveness of various machine learning techniques, face spoofing assaults can be recognized. It also compares the top classifier to earlier studies in this area. Several attack methods—including replay and print attacks—are deployed to create test data sets for the suggested strategy [37].

3.1. Proposed Face Spoof Attack Recognition Model

The proposed Face Spoof Recognition Scheme is presented in Fig 2.

It provides an example of how to identify and categorize parody images step-by-step. The first step is to show the system an image. The binary format conversion of this image file allows for the extraction and selection of particular features. After that, the components are categorized based on the criterion. The face spoof recognition module then implements a classifier to recognize the fake face's characteristics [38]. The algorithm decides which qualities to use depending on how they are used. The worth of each feature's content is assessed using a learning module, and the element is automatically selected. The parts are then ranked in order of importance.

The instruction module manages this on its own as well [1]. The classifiers look at each of these sets of features separately to look for spoofing attempts. The classification accuracy of several machine learning techniques is examined in the suggested classification model for the face spoof attack recognition system. The results are contrasted with those of additional deep learning methods like CNN and RNN and the five machine learning algorithms employed in the face spoof recognition system: KNN, NB, DT, RF, and SVM. A confusion matrix assesses O.C.ROC analysis and the categorization findings. The automatic prioritization of spoof assaults is examined using the confusion matrix.
3.2. Proposed Architecture of the Classification Algorithm for Face Spoof Recognition

The size of the pooling layer and the filters matters more when building a CNN model than the number of convolution layers. As a result, several tests were conducted using various combinations of filter widths and convolution depths. The intricacy of the CNN architecture was considered throughout the study. A Python program determines which CNN architecture performs best for a certain input picture size. Learning how to operate such a complex layout takes a lot of time. This is because a single convolution layer cannot provide sufficient information. However, adding more than three convolution layers makes the model more complex and increases training time. Scherer, Müller, and Behnke should note that while comparing models with two and three convolution layers, the performance was highest when there were as many pooling layers as was realistically achievable [39]. The dropout value is employed in a neural network simulation to use the model in a larger variety of circumstances. A hidden layer after the single layer runs the length of the design and is linked throughout. Starting with the first planning stages, the total number of nodes in the hidden layer is considered for optimized results. The model that was shown has a similar structure but distinct components. Convolutional processing began with only three layers. Since the concept is so intricate, you are limited to using a pool 2 by 2 by 2.

We utilized patches of many sizes to determine the ideal method of erecting the structure. Various patch sizes were employed to define "moderate patch size" and begin a conversation about it. A filter with these dimensions may have three convolutional layers on its surface. The models became more complex as the filter sizes increased. In the end, it was discovered that a pooling layer should be no larger than 2 by 2 by 2 in size. Numerous patch sizes were tested throughout training, but 24 32 by 32 proved the ideal arrangement. Details of experimentation performed to find a suitable architecture for maximized accuracy are described in Table 1. Table 2 shows the evaluation of the Sensitivity of Models. The proposed method is based on classical methods of deep machine learning. However, the focus was to find architecture that can improve face spoof detection accuracy. After extensive experimentation, one such optimized architecture, discussed below in Table 1, resulted in improved accuracy.

Table 1: Specifications of the Various Experimentation Models.
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Experiment1</th>
<th>Experiment2</th>
<th>Experiment3</th>
<th>Experiment4</th>
<th>Experiment5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of Patch</td>
<td>12 x 24 x 24</td>
<td>18 x 30 x 30</td>
<td>24 x 36 x 36</td>
<td>28 x 42 x 42</td>
<td>36 x 42 x 42</td>
</tr>
<tr>
<td>Conv. Layer 1</td>
<td>3, 5, 5 (Same for All Experiments)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max Pooling</td>
<td>3, 3, 3 (Same for All Experiments)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drop out</td>
<td>0.15</td>
<td>0.15</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>Conv. Layer 2</td>
<td>3, 5, 5</td>
<td>3, 6, 6</td>
<td>4, 6, 6</td>
<td>3, 5, 5</td>
<td>3, 5, 5</td>
</tr>
<tr>
<td>Max Pooling 2</td>
<td>2, 2, 2 (Same for all experiments)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drop out for Max Pooling 2</td>
<td>0.1</td>
<td>0.150</td>
<td>0.150</td>
<td>0.150</td>
<td>0.150</td>
</tr>
<tr>
<td>FCN 1 nodes</td>
<td>100</td>
<td>160</td>
<td>240</td>
<td>260</td>
<td>360</td>
</tr>
<tr>
<td>FCN 2 Nodes</td>
<td>2 (Same for All experiments)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type of Activation Function</td>
<td>Soft max</td>
<td>Sigmoid</td>
<td>Sigmoid</td>
<td>Sigmoid</td>
<td>Sigmoid</td>
</tr>
</tbody>
</table>

**Table 2: Evaluation of Sensitivity of Models.**

<table>
<thead>
<tr>
<th>Experimented Architecture</th>
<th>Sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimental Model 1</td>
<td>0.670</td>
</tr>
<tr>
<td>Experimental Model 2</td>
<td>0.689</td>
</tr>
<tr>
<td>Experimental Model 3</td>
<td>0.745</td>
</tr>
<tr>
<td>Experimental Model 4</td>
<td>0.737</td>
</tr>
<tr>
<td>Experimental Model 5</td>
<td>0.802</td>
</tr>
</tbody>
</table>

The sensitivity of the experimental models is evaluated. Then, the model with the highest sensitivity is proposed as a better-performing model for this work to test over multiple datasets. It is observed that the 5th experimental model achieved better sensitivity, which is further evaluated over multiple datasets for face spoof classification.

The face spoofing recognition module's architecture includes ways for categorizing face spoofing attacks. This type of learning module uses statistical reasoning to attempt to determine whether a second image is real, live, or whether it's a malicious spoof attack. Typical systems don't have a process code. Instead, if-then reasoning is used. Our research does not employ machine learning when creating a
learning module suitable for expert systems. We instead use if-then rules. The system decides how to classify items using a sequential approach. Finding the original image on the user's device is the first step. Right now, the photo file reader is helpful. RGB image files, like JPG or PNG, are the building blocks of picture files. After that, the image goes through preliminary processing. Any unnecessary data is deleted after the image file has been converted to binary format and normalized. This image is sent to the machine learning module so that pre-selected features can be obtained and processed [39]. Algorithms automatically conduct classifications once components are removed. The classification process that machine learning techniques employ to spot fakes is covered in the next section. Fig 3 presents the proposed Face Spoof Recognition Technique. Finally, the results of the classification of various algorithms are compared. For this purpose, a different module is developed.

This system consists of various layers. The first layer is the object layer responsible for obtaining images from the dataset. The second layer is the data acquisition layer, which converts the input image into a raw binary image and sends it for pre-processing. During the pre-processing, the image is looked for various tasks, including alignment adjustment, spikes, handling of missing values, and removal of white noise. After that, CNN input layers start working to find correlation inference at the hidden layer based on learned weights and results in the classification of whether the input image is real or a spoof. The classifier selected in this system is CNN after the recommendation of extensive testing performed using the comparative module given in Fig 4. In the end, performance evaluation is performed by the performance evaluation layer, and a confusion matrix is generated along with the output sent to the user.

Fig 4 describes the working of the module developed to run and compare various classifiers to recommend a classifier with maximum accuracy and better results. The module takes input images from various data sets and automatically prioritizes features. After prioritizing features, the module reduces features by only focusing on prioritized features. The module then sequentially runs all the classifiers to find their accuracy and then provides analysis reports for all classifiers. Based on the recommendation of this module, CNN was selected as a suitable classifier to be used in the proposed system mentioned in Fig 3.

3.3. Data Set for Evaluation of Algorithm

Images from the Kaggle data repository made up the data set for this research. Using Matlab, students are taught how to identify fake photos in the learning module. Teachers and students can use the data set for instructional purposes because it is available online.

1-Thirty-five users have been identified in the MSU MFSD database as face spoofing victims. In this dataset, 15 of the 35 people are trained, and the other 20 are used for testing. High-resolution video attacks, replay video attacks on mobile devices, and A3 paper pictures were all used to spoof the client's biometrics.
2. You can look at data collection about the Replay attack at the IDIAP research institute. It has about 1300 videos on 50 genuine and fake topics. A MacBook laptop was used to record the videos in two lighting scenarios: good and bad. To learn, develop, and test, 15-person groups, 15-person groups, and 20-person groups view videos. Whether a machine or a person handles the fake media, print and replay assaults are split into two groups in the database [40].

3. The IDIAP research center offers details about the Replay attack. Over 1300 real and fake videos covering 50 different topics are available. Two lighting conditions—one with good lighting and the other with bad lighting—were captured using a MacBook. People watch movies in groups of 15, 15, and 20 to learn, improve, and get evaluated. The database divides print and replay attacks into categories depending on whether a machine or a person carried them out.

4. Results and Discussion

4.1. Experimentation Scenario

The Matlab simulator was used to produce this situation. To complete this simulation, Matlab 2018R2 was used. The dataset assesses how various classification methods performed on simulated data are healthy. The forecast's accuracy is the most important thing to consider, though. The system's analysis and current data are shown below:

- The simulation tool used is Matlab2018R2,
- The algorithm for simulation is N.B., K.N.N., SVM, D.T., CNN, R.N.N.,
- In the dataset, the number of instances used is 9000,
- The selection of features is automatic,
- Our used dataset is IDIAP, MSU-USSA,
- CPU used is Intel core I-10,
- The used memory is 16 GB.

4.2. Implementation of Proposed Model

The classification model is described in detail in Fig 3. This model is implemented as a part of the learning module. The model learns from data kept in a database—the NB, SVM, KNN, DT, CNN, and. RNN machine learning algorithms are used sequentially in this model to produce the best results. Implementing the NB, SVM, and K.N.N.KNN algorithms automatically selects features. Some criteria for assessing these processes' effectiveness include recall, precision, and accuracy.

4.3. Parameter Setting for Classification Algorithms

Different settings are used for algorithms during the execution of the classification algorithm. The grounds are described in the following.
4.3.1. Parameter Setting for K-Nearest Neighbours

Twenty neighbours were designated for the execution. The kind of Linear metric was also chosen. Equal weight was distributed among the neighbours.

4.3.2. Parameter Setting for SVM.

C-SVM is used to evaluate this study. The SVM learning component uses RBF kernels. C talks about optimization and how to avoid classifying the training instance incorrectly. C has been set to 1.0 for this simulation. The tolerance has been set at 0.001 percent after 100 simulation runs. It is noteworthy that the expr. Value is also set to \((-1.01 |x-y|2)\). The optimization function chooses the tighter training margins of the hyper plan when C is high. The search for hyperplanes that are incredibly far apart from one another is made possible by a lower value of C during Learning.

4.4. Classification Error Rate

The error rate was calculated by combining test error and training error and multiplying them with their weighted averages. For the test dataset, 0.632 is selected as the classification error rate, while for the training data set, the classification error weight is selected to be 0.368. The quadratic error during training is given less weight during training. The formula used to calculate the error is presented below.

\[
err = 0.632 \times e_{\text{test instances}} + 0.368 \times e_{\text{training instances}}
\]  

To the greatest extent possible, the learning error is minimized by calculating "iteration by iteration." The quadratic error predictions are averaged over the training period. The graph below shows the actual quadratic error values. The previous cycle's results showed that the classifier had little effect on deep Learning, with a quadratic error of 0.01981. The Convolutional Neural Network is used in the "iterative" phase to show how Learning functions at its most basic level [44]. The training error function's capacity for Learning impacts a student's ultimate grade. A system must have fewer errors for it to work correctly. The difference between the intended and actual results is the mean square error. Training error measures the disparity between a system's planned and essential functions.

A mathematical model can be used to show how this differentiation differs. The training error is checked during the training period to ensure that the performance of identifying face spoofs complies with the set standards. The model makes a mistake after it has been trained and used on a batch of data for the first time. This typically happens when a model has been applied several times to the same data set and yields inconsistent or false results. When a trained model uses data different from the dataset it was trained on and unfamiliar with, it makes a test error. Both errors can be measured using the training error, which can be applied to testing and training. The training error shows the average number of mistakes made by the subgroups in the data set utilized for training and testing. The training error values drop with
each iteration until they get close to 0.01985, an excellent sign of an estimator's accuracy. As the system continues to learn, as seen in Table 1, the training error rate is low. Fig 5 shows the CNN Training Error.

The error rate decreases linearly with the number of training iterations. The neural network’s lowest error has been reduced to 0.01985 after 525 training iterations.

4.5. Experimentation and Analysis

The effectiveness of different machine learning techniques, including SVM, NB, and KNN, was evaluated and compared during experimentation—a graph showing the number of TP and FP factors are provided to assess performance. RMSE analysis and other performance comparison approaches understand the outcomes and real-time data well. The descriptions of these significant performance metrics are provided in further detail below. The image file data set was used as input for the suggested use of machine learning methods. People look into how network categorization changes when the fitness function is altered. Fig 6 shows an example of Instances from the USSR dataset: Row 1 and Row 2 contain real faces, and Row 3 and Row 4 contain real-time Spoofed images of different cameras. Fig 6 shows the USSR dataset: Row 1 and Row 2 contain real faces, and Row 3 and 4 contain Spoofed images of different cameras

4.6. Training Parameters for Deep Learning Model

The face spoof images are resized to 224x224 pixels before being utilized for training the model. The model is trained using conventional stochastic gradient descent. Using stochastic gradient descent, it is simple to determine which expectation and observation best match the data. SGD is a method for choosing the best values for the parameters of a machine learning model. The model includes many face photos. When momentum is set to 0.90, the weight decay rate is set to 0.0005, and the learning rate decay rate is set to 10^-7. Is 0.001, an extra key parameter, also used for the regularization parameters of the loss function, so long as it is a power of one?

4.7. Feature Extraction

The features of the images, including texture patterns, color distortions, and geometric distortions, serve as the basis for these algorithms’ inputs.

4.7.1. Image Texture Patterns

The texture and color of the fake photo paper's surface are different. Materials with recurrent patterns are recognized using faces. Last but not least, the binary feature patterns receive 479 more dimensions.

4.7.2. Image Color

To compare and match textures and photos, image color is also used. This feature is very helpful for classifying presentation attacks. This feature compares color spaces in a single feature vector, including HSV and YCbCr [41].
4.7.3. Distortion

The term "distortion" can describe a wide range of occurrences, such as a change in shape, the perception of color, or the way objects fit together. The photograph was discovered to exhibit a specular reflection by spotting a fake. To calculate the distortion value, a 121 feature vector is used [43].

4.7.4. Entropy Loss

Convolutional Neural Network stands out for their ability to differentiate entropy during deep Learning. The cross-entropy loss algorithm helps Alex Net pick up on this feature during training. On the other hand, transfer learning uses cross-entropy loss to train on a more complex data set (Fine-tuned CNN) [42]. The confusion matrix achieved after the evaluation of each algorithm is given in Table 3, while a comparison of different algorithms and their accuracy is presented in Table 4.

Table 3: Confusion Matrix of Various Algorithms.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>TN</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>831</td>
<td>136</td>
<td>103</td>
<td>545</td>
</tr>
<tr>
<td>KNN</td>
<td>791</td>
<td>172</td>
<td>214</td>
<td>438</td>
</tr>
<tr>
<td>NB</td>
<td>787</td>
<td>156</td>
<td>224</td>
<td>448</td>
</tr>
<tr>
<td>CNN</td>
<td>872</td>
<td>113</td>
<td>90</td>
<td>540</td>
</tr>
<tr>
<td>DT</td>
<td>733</td>
<td>159</td>
<td>353</td>
<td>370</td>
</tr>
<tr>
<td>RNN</td>
<td>845</td>
<td>155</td>
<td>97</td>
<td>518</td>
</tr>
</tbody>
</table>

Table 4: Comparison of different algorithms and their accuracy.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Print Attack</th>
<th>Replay Attack</th>
<th>#3D Mask Attack</th>
<th>Average Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>91.66</td>
<td>84.30</td>
<td>79.60</td>
<td>85.19</td>
</tr>
<tr>
<td>KNN</td>
<td>89.62</td>
<td>70.10</td>
<td>68.50</td>
<td>76.07</td>
</tr>
<tr>
<td>NB</td>
<td>77.40</td>
<td>79.70</td>
<td>72.30</td>
<td>76.47</td>
</tr>
<tr>
<td>CNN</td>
<td>84.67</td>
<td>88.60</td>
<td>89.00</td>
<td>87.42</td>
</tr>
<tr>
<td>DT</td>
<td>76.50</td>
<td>65.45</td>
<td>63.01</td>
<td>68.32</td>
</tr>
<tr>
<td>RNN</td>
<td>88.76</td>
<td>84.60</td>
<td>79.80</td>
<td>84.39</td>
</tr>
</tbody>
</table>

The comparative results of the average accuracy of various algorithms are shown in Fig 9. In contrast, the accuracy of face spoof detection of various algorithms against various types of attacks is shown in
4.8. Comparison of Equal Error Rate (EER.)

The performance of a biometric system is typically evaluated in verification tasks using an equal error rate. All ROC and DET curves have the same absolute acceptance and false rejection rates. The accuracy of a biometric system rises as the rate of equal errors falls. A comparison of Equal Error Rate is presented in Table 5 and Fig 11.

A comparison of the AUC Score achieved by various classifiers is given in Table 6 & Fig 12.

Table 6: AUC Score of Various Classifiers.

<table>
<thead>
<tr>
<th>Sr No.</th>
<th>Classifier Name</th>
<th>AUC Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CNN</td>
<td>0.9245</td>
</tr>
<tr>
<td>2</td>
<td>KNN</td>
<td>0.83</td>
</tr>
<tr>
<td>3</td>
<td>RNN</td>
<td>0.916</td>
</tr>
<tr>
<td>4</td>
<td>DT</td>
<td>0.67</td>
</tr>
<tr>
<td>5</td>
<td>NB</td>
<td>0.8</td>
</tr>
<tr>
<td>6</td>
<td>SVM</td>
<td>0.9049</td>
</tr>
</tbody>
</table>

Fig 11 presents the comparison between equal error rates achieved after the evaluation of various algorithms. The algorithm's sensitivity and specificity are shown via ROC graphs. Specificity is shown on the x-axis, while algorithm sensitivity is depicted on the y-axis. An AUC value of 0.5 is generally considered a poor result since it does not accurately reflect the algorithm's capacity to identify a particular decision threshold. Moreover, an AUC value of 0.7-0.8 is adequate for its detection to be almost correct. The results describe that CNN, RNN, and SVM achieved maximum. AUC score (0.9245, 0.916, 0.9049 respectively) with its curve tilting towards the top left corner more than other algorithms. Results describe the accuracy of the. DT is below all the other algorithms. It reported an average. AUC score of 0.67, which also represents its low accuracy value of detection for the spoof attack. On the other hand, Naïve Bayes. AUC is averaged as 0.80, representing its acceptable value with insufficient accuracy required to be used as an expert system classifier. AUC score is. KNN is reported to be between 0.8 and 0.9 with an average value of 0.83, which shows it has good prediction accuracy for heart disease diagnosis. However, Convolutional Neural network implementation was reportedly achieved. AUC score of 0.924, showing its ability to face spoof attacks with near-real estimates.

5. Comparison of our Proposed Model with Other Methods

Our proposed model is compared with other state-of-the-art methods, particularly SOTA methods. The comparison is given in the following Table 7. The comparison is made concerning average precision, F1
score, and classification accuracy.

This study uses automated feature extraction, which can only pick up a finite number of features. Picking more features may increase or decrease the model's accuracy and efficiency.

Table 7: Comparison of Other Classifier's performance with our proposed model.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Precision</td>
<td>0.607</td>
<td>0.718</td>
<td>0.735</td>
<td>0.783</td>
<td>0.904</td>
<td>0.8742</td>
</tr>
<tr>
<td>F1 Score</td>
<td>0.505</td>
<td>0.601</td>
<td>0.764</td>
<td>0.759</td>
<td>0.773</td>
<td>0.8957</td>
</tr>
<tr>
<td>No. of Model Parameters</td>
<td>3.06 M</td>
<td>28.4 M</td>
<td>24.0 M</td>
<td>8.9 M</td>
<td>8.98 M</td>
<td>9.71 M</td>
</tr>
<tr>
<td>Model Complexity (GFlops)</td>
<td>1.14</td>
<td>908.1</td>
<td>61.2</td>
<td>11.27</td>
<td>11.26</td>
<td>12.5</td>
</tr>
</tbody>
</table>

6. Conclusion

This study uses machine learning and deep Learning to examine face spoofing attacks on machine learning systems. To offer clients improved security services, face spoofing must be rectified in machine learning-based expert systems that use biometrics and unique Face photos. Machine learning-based classifiers have been created and validated to provide more biometric security than rule-based detection or decision-making based on colour distortion. Due to its high classification precision and low prediction error, this reliable biometric security solution enables service providers to detect face spoofing threats in real time. Convolutional neural networks exceed their competitors' precision in classification, prediction, and AUC output. RNN needs a more extensive training set to learn classification thresholds with more accuracy, but it can also be used to boost the accuracy of deep learning algorithms. Due to its poor accuracy and many false positives, DT was also the least reliable. A face spoof classification system can be set up more quickly because of the Convolutional Neural Network-based classification module's compatibility with the testing and validation modules. It is observed that the convolutional neural network improves the accuracy and prevision of classification with more AUC output when compared with others.
On the other hand, RNN can also be used for improved accuracy in deep learning algorithms, but it requires more training data sets for learning classification thresholds with more accuracy. Moreover, DT resulted in the least accuracy with the highest values of false positives. Therefore, the CNN-based classification module is more supportive and helpful in implementing the face spoof classification system in collaboration with the testing and validation module.

7. Future Work

This work is based on the experimental evaluation of the proposed model using the MSU, MSFD, and IDIAP data sets. In the future, we aim to extend this over other datasets with different deep-learning algorithms. It is also evident that a real-time testbed implementation of the proposed method would help improve the system's performance. Additionally, this study uses automated feature extraction, which can only pick up a finite number of features. We aim to train our proposed model for various individual features in the future.

Declarations

Ethical Approval: Not applicable

Availability of data and materials: Dataset is publically available.

Acknowledgment: Not applicable

Funding Statement: No Extra Funding

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References


27. Atoum, Y. et al. Face anti-spoofing using the patch and depth-based CNNs. in 2017 IEEE International Joint Conference on Biometrics (IJCB), IEEE, 2017


49. Zhao, YuanZhang, and ShengLing Geng. "Object detection of face mask recognition based on improved faster RCNN." In 2nd International Conference on Computer Vision, Image, and Deep

Table
Table 5 is not available with this version.

Figures

Figure 1
Examples of Face Spoof attacks represent two spoofing attempts where two different real faces (left) apply makeup (middle) to resemble the same target identity (right).

Figure 2

Generic Method of Face Spoof Recognition.
Figure 3

Proposed Classification Algorithm of Face Spoof Recognition.

Figure 4

Working of Module for Comparative analysis of various classifiers.
Figure 5

CNN Training Error.

Figure 6
Example of Instances from the USSA dataset: Row 1 and Row 2 contain real faces, and Row 3 and Row 4 contain real-time Spoofed images of different cameras. Fig 7 shows an example of Instances from the IDIAP dataset. Row 1 contains the Images of the IDIAP-Replay attack, And Row 2 includes the adverse images.

Figure 7

Example of Instances from IDIAP dataset. Row 1 contains the Images of the IDIAP-Replay attack, And Row 2 includes the adverse images. For example, images of different cameras used by the MSU datasets are presented in Fig 8.

Figure 8
Images of different cameras used in the MSU dataset.

**Figure 9**

Comparison of Accuracy.

**Figure 10**

Comparison of Accuracy against Various Face Spoof Attacks.
Figure 11

Comparison of Equal Error Rate.

Figure 12

ROC Analysis of various Algorithms.